SC24/WG7 N1597

ISO/IEC Draft International Standard (DIS) 12087-5

INFORMATION TECHNOLOGY

COMPUTER GRAPHICS AND IMAGE PROCESSING

IMAGE PROCESSING AND INTERCHANGE

FUNCTIONAL SPECIFICATION

Part 5:

BASIC IMAGE INTERCHANGE FORMAT

11 July 1997

EDITOR'S REMARKS

The project editor for this document is:

Robert W. Garneau, Ph.D.
TASC, Inc.
55 Walkers Brook Drive
Reading, Ma. 01867.3297
USA

phone: +1 617 942 2000
fax: +1 617 942 0636
E-Mail: rwgarneau@tasc.com

This document is the Part 5 draft supplement to the ISO document titled:

Information Technology
Computer Graphics and Image Processing
Image Processing and Interchange
Functional Specification
Part 5:

Basic Image Interchange Format

Editor's Note: The editor would like to thank the Technical Experts that have contributed to the development of this document since I became editor at the WG7 meeting in Japan. By both direct submission and contributions made during editing meetings, these experts have made the development of this document possible. This has truly been a team effort. Thank you.

Forward

The International Standard for Image Processing and Interchange, developed by ISO/IEC JTC1 SC24, 12087 includes 4 parts:

Part 1: Common Architecture for Imaging;
Part 2: Programmer's Imaging Kernel System Application Program Interface;
Part 3: Image Interchange Facility;
Part 5: Basic Image Interchange Format.

The title of this International Standard shall be:

Computer Graphics and Image Processing - Image Processing and Interchange
Part 5: Basic Image Interchange Format

Annexes A, B and C form an integral part of ISO/IEC 12087-5. Annexes D and E are for information only.

Contents

List of Figures

List of Tables

1 Scope and field of application

This document, Part 5 of the International Standard for Image Processing and Interchange (IPI), establishes the specification of the Basic Image Interchange Format (BIIF) part of the standard. BIIF is a standard developed to provide a foundation for interoperability in the interchange of imagery and imagery-related data among applications. This standard provides a detailed description of the overall structure of the format, as well as specification of the valid data and format for all fields defined with BIIF. Annex C contains a model profile in tables to assist in profile development.

As part of the 12087 family of image processing and interchange standards, BIIF conforms to the architectural and data object specifications of 12087-1, the Common Architecture for Imaging. BIIF supports a profiling scheme that is a combination of the approaches taken for 12087-2 (PIKS), 10918 (JPEG), 8632 (CGM), and 9973 (The Procedures for Registration of Graphical Items). It is intended that profiles of the BIIF will be established as an International Standardised Profiles (ISP) through the normal ISO processes (ISO/IEC TR 10000).

The scope and field of application of BIIF includes the capability to perpetuate a proven interchange capability in support of commercial and government imagery, Programmer's Imaging Kernel System Data, and other imagery technology domains in that priority order.

BIIF provides a data format container for image, symbol, and text, along with a mechanism for including image-related support data.

BIIF satisfies the following requirements:

- Provides a means whereby diverse applications can share imagery and associated information.
- Allows an application to exchange comprehensive information to users with diverse needs or capabilities, allowing each user to select only those data items that correspond to their needs and capabilities.
- Minimizes preprocessing and postprocessing of data.
- Minimizes formatting overhead, particularly for those applications exchanging only a small amount of data and for bandwidth-limited systems.
- Provides a mechanism (Transportable File Structure, TFS) to interchange PIKS image and image-related objects
- Provides extensibility to accommodate future data, including objects.

When the extensibility of BIIF, or the inherent constraints of the structured format of BIIF, do not meet the needs of a more complex application, the concepts and features of 12087-3 (IIF) should be considered as a more appropriate method of image interchange. For example, the ability to support complex combinations of heterogeneous pixel types, self defining pixel structures, or abstract structures can be done with IIF.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard should investigate any recent editions of the standards listed below.

The following are normative references to this Standard.

ISO/IEC 8601:1988 Data elements and interchange formats - Information interchange - Representation of dates and times - Technical Corrigendum 1 : 1991

ISO/IEC 4873:1991(E) Information technology - ISO 8-bit code for information interchange - Structure and rules for implementation

ISO/IEC 2022:1986 Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques

ISO/IEC 646:1991 Information technology: ISO 7-bit coded character set for information interchange

ISO/IEC 10646:1993 Information Technology - Universal Multiple-Octet Coded Character Set (UCS) Part 1 : Architecture and Basic Multilingual Plane

ISO/IEC 10646:1993 Information Technology - Universal Multiple-Octet Coded Character Set (UCS) Part 1 : Architecture and Basic Multilingual Plane - Amendment 2: UCS Transformation Format 8 (UTF-8)

ISO/IEC 10918-1:1994 Information technology - Digital compression and coding of continuous-tone still images : Requirements and guidelines

ISO/IEC 10918-2:1995 Information technology - Digital compression and coding of continuous-tone still images : Compliance testing

ISO/IEC 10918-3:DIS Information Technology; Digital Compression and Coding of Continuous-Tone Still Images; Part 1: Extensions

ISO/IEC 10918-4:CD Information Technology; Digital Compression and Coding of Continuous-Tone Still Images: Part 4; Registration Procedures for JPEG Profile, APPn Marker, and SPIFF Profile ID Marker

ISO/IEC 8632-1:1994 Information technology - Computer graphics - Metafile for the storage and transfer of picture description information - Part 1: Functional specification Amendment 1:1994 to ISO/IEC 8632-1:1992 Rules for profiles Amendment 2: 1995 to ISO/IEC 8632-1:1992 Application structuring extensions

ISO/IEC 8632-2:1994 Information technology - Computer graphics - Metafile for the storage and transfer of picture description information - Part 2: Character encoding Amendment 1:1994 to ISO/IEC 8632-1:1992 Rules for profiles Amendment 2: 1995 to ISO/IEC 8632-1:1992 Application structuring extensions

ISO/IEC 8632-3:1994 Information technology - Computer graphics - Metafile for the storage and transfer of picture description information - Part 3: Binary encoding Amendment 1:1994 to **ISO/IEC 8632-1:1992** Rules for profiles Amendment 2:1995 to **ISO/IEC 8632-1:1992** Application structuring extensions

ISO/IEC 8632-4:1994 Information technology - Computer graphics - Metafile for the storage and transfer of picture description information - Part 4: Clear text coding Amendment 1:1994 to ISO/IEC 8632-4:1992 Metafile for the storage and transfer of picture description information Amendment 2: 1995 to ISO/IEC 8632-4:1992 Application structuring extensions

ISO/IEC 9973:1994-12-15 1 st Edition, Procedures for Registration of Graphical Items

ISO/IEC 11072:1993 Information technology - Computer graphics - Computer Graphics Reference Model

ISO/IEC 12087-1:1995 Information technology - Computer graphics and image processing - Image processing and Interchange--Functional specification Part 1: Common architecture for imaging

ISO/IEC 12087-2:1994 Information technology - Computer graphics and image processing - Image processing and Interchange--Functional specification Part 2: Programmer's imaging kernel system application programme interface

ISO/IEC 12087-3:1995 Information technology - Computer graphics and image processing - Image processing and Interchange--Functional specification Part 3: Image Interchange Facility (IIF)

ISO/IEC TR10000-1:1994 Information Technology - Framework and Taxonomy of International Standardised Profiles Part I Framework 3nd Edition 1994-12-15

ITU-T T.4 (1993:03) Standardisation of Group 3 Facsimile Apparatus for Document Transmission, AMD2 08/95

3 Definitions, symbols and abbreviations

3.1 Definitions:

For the purpose of ISO/IEC 12087, the following definitions apply.

3.1.1. **Attachment Level** - A field value of a segment that indicates the display level of the segment to which it is attached. It provides a way to associate images and symbols as a group for the purpose of moving, rotating or displaying.

3.1.2. **Annotational Text** - Also referred to as Symbol Text. It is text placed on an image as a graphic symbol to provide a textual overlay to the image. See ISO/IEC 8632:1992

3.1.3. **Band -** One of the two-dimensional (row/column) arrays of pixel sample values that comprise an image. For the basic use of BIIF, the band values are homogeneous data types for each band. In the case of monochrome or indexed colour images (single, 2 dimensional array of pixel values with possible look-up-tables), the image array consists of one band. In the case of RGB images (three 2-dimensional arrays of pixel values; 8 bits each of Red, Green and Blue values for each pixel), the image consists of three bands. When images need to be represented using bands with heterogeneous array structure or data types (e.g. two bands with integer data type and one band with a real data type), the image representation may be defined using a PIKS object in a TFS Data Extension Segment (DES). The TFS PIKS object defines the data structure of the values in the image data field of the image segment.

3.1.4. **Basic Character Set (BCS)** - A subset of ISO/IEC 10646 character set which is represented by the UTF-8 form and used in headers and subheaders.

3.1.5. **Basic Character Set-Alphanumeric (BCS-A)** - A subset of the Basic Character Set. The range of allowable characters consists of space through tilde (single octets with values ranging from 20 to 7E) from the Basic Latin Collection.

3.1.6. **Basic Character Set-Numeric (BCS-N)** -A subset of the Basic Character Set which consists of the digits ‘0’ through ‘9’, ‘plus sign’, ‘minus sign’, ‘decimal point’, and ‘slash’.

3.1.7. **Basic Multilingual Plane** - The Basic Multilingual Plane (BMP) is defined as group 00 of plane 00. The BMP includes characters in general use in alphabetic, syllabic, and ideographic scripts together with various symbols and digits. See ISO/IEC 10646:1993.

3.1.8. **Block** -A rectangular array of pixel values which is a subset of an image. An image consists of the union of one or more non-overlapping blocks.

3.1.9. **Byte** - A byte is defined as equivalent to an octet.

3.1.10. **Common Coordinate System** - A two dimensional coordinate space which is common for determining the placement and orientation of displayable data types within a specific BIIF file and among correlated BIIF files which comprise an integrated product.

3.1.11. **Conditional** - An adjective applied to data fields whose existence depends on the value of the designated Required field preceding the Conditional field.

3.1.12. **Coordinated Universal Time (UTC)** - The time scale maintained by the Bureau International de L'Heure (International Time Bureau) that forms the basis of a coordinated dissemination of standard frequencies and time signals. The time zone indicator for Greenwich Mean Time (GMT). UTC is equivalent to the mean solar time at the prime meridian at Greenwich.

3.1.13. **DES** - Data Extension Segment is a construct used to encapsulate different data types. The DES structure is discussed in Subclause 4.2.8.3.

3.1.14. **Displayable** - Information that can be exhibited in visual form.

3.1.15. **Display Level** -A field value of a segment that denotes the order in which the segments (images and symbols) are “stacked”. The Display Level order is independent of the data sequence order in this format.

3.1.16. **Field** - Logically primitive item of data, sometimes referred to as an attribute.

3.1.17. **Image** - A representation of physical visualization, for example, a picture. An image is the computer (digital) representation of a picture. An image is comprised of discrete picture elements called pixels structured in an orderly fashion consisting of pixel value arrays formatted using bands and blocks.

3.1.18. **International Standardised Profile** - An internationally agreed-to, harmonized document which identifies a standard or group of standards together with options and parameters necessary to accomplish a function or set of functions.

3.1.19. **Look-Up Table** - A collection of values used for translating image samples from one value to another. The current sample value is used as an index into the look-up table(s); therefore, the number of entries in each look-up table for a single bit per pixel image would contain two entries, and each look-up table for an 8-bit per pixel image would contain 256 entries. Multiple look-up tables allow for the translation of a 1-vector pixel value to an n-vector pixel value.

3.1.20. **Non-blank** - Non-blank indicates that the field cannot be filled entirely by the BCS-A space character (0x20). It may contains pace characters when included with other characters

3.1.21. **Octet** - An octet is defined as 8 bits.

3.1.22. **Pad Pixel** - A pixel with sample values that have no significant meaning to the image. Pad pixels are used with block images when either the number of pixel rows in an image is not an integer multiple of the desired number of vertical image blocks, or when the number of pixel columns in an image is not an integer multiple of the desired number of horizontal image blocks.

3.1.23. **Pad Pixel Mask** - A data structure which identifies recorded/transmitted image blocks which contain pad pixels. The pad pixel mask allows applications to identify image blocks which require special interpretation due to pad pixel content.

3.1.24. **Pixel** - An abbreviation for the term “picture element”.

3.1.25. **Profile** - A set of one or more base standards, and where applicable, the identification of chosen classes, subsets, options, and parameters of those base standards, necessary for accomplishing a particular function.

3.1.26. **Profile Variant** - A field within this basic standard that is allowed to be defined by a profile for its structure and intent (content). An element or an attribute that is allowed to differ between profiles.

3.1.27. **Required** -An adjective applied to data fields that must be present and filled with valid data or default data.

3.1.28. **RES** - Reserved Extension Segments are extension support structures provided in the BIIF, each made up of a subheader and corresponding data field. The RES structure is discussed in Subclause 4.2.8.4.

3.1.29. **Sample** - One element in the image array that comprises an attribute of the image. In BIIF, a sample (pixel vector value) is indexed according to the row and column of the array where it appears.

3.1.30. **SAR** - Image obtained from a synthetic aperture radar.

3.1.31. **SARIQ** - Radio hologram (initial phase information) from a synthetic aperture radar.

3.1.32. **Segment** - An instance of a data type that is contained in a BIIF file. A segment is comprised of a subheader and associated data. E. g. an image subheader together with image data comprises an image segment.

3.1.33. **Symbol** - A pictorial element that may be aligned with a point in or adjacent to an image to provide graphical markings and/or textual labels.

3.1.34. **Symbol Text** - Text placed on or adjacent to an image as a graphic symbol. See ISO/IEC 8632-1:1992 Computer Graphics Metafile (CGM) for a full description of Symbol Text.

3.1.35. **Tagged Record Extension (TRE)** - A means to provide additional attributes about standard data segments not contained in the standard header or sub-header fields.

3.1.36. **TFS** - Transportable File Structure allows for configuration, data request, commands and PIKS object data to be stored in hierarchy order with Metadata associated for each level. TFSs are defined and demonstrated in Annex A.

3.1.37. **Transparent Pixel** - A pixel whose sample values must be interpreted for display such that the pixel does not obscure the display of any underlying pixel.

3.1.38. **YCbCr** - Technique for specifying colour images where Y-Brightness of signal, Cb=Chrominance (blue), Cr=Chrominance (red)

3.2 Abbreviations:

AL Attachment Level
BCS Basic Character Set
BCS-A Basic Character Set - Alphanumeric
BCS-N Basic Character Set - Numeric
BMP Basic Multilingual Plane
C Conditional
CCS Common Coordinate System
CGM Computer Graphics Metafile
CS Character String
DES Data Extension Segments(s)
DL Display Level
JPEG Joint Photographic Experts Group
LSB Least Significant Bit
LUT Look-Up Table
MSB Most Significant Bit
NBPC Number of Blocks Per Column
NBPR Number of Blocks Per Row
PVTYPE Pixel Value Type
RES Reserved Extension Segment (s)
RGB Red, Green, Blue
YCbCr Y-Brightness of signal, Cb=Chrominance (blue), Cr=Chrominance (red)
TRE Tagged Record Extension
TFS Transportable File Structure
UCS Universal Multi-Octet Coded Character Set
UCS-2 UCS Two Octet Form
UCS-4 UCS Four Octet Form
UTC Coordinated Universal Time
UTF-8 UCS Transformation Format 8

4 Basic image interchange format (BIIF) specification

4.1 Format overview

Imagery applications use multiple types of systems for the exchange, storage, and processing of images and associated imagery data. The format used in one application is likely to be incompatible with formats used in other applications. Since each application may use a unique, internal data representation, a common format for interchange of information across applications is needed for interoperability of systems within and across applications. This clause defines the Basic Image Interchange Format (BIIF) specification to provide a common basis for storage and interchange of images and associated data among existing and future applications. BIIF supports interoperability by providing a data format for shared imagery and an interchange format for images and associated imagery data.

In BIIF, data interchange between disparate systems is potentially enabled by a translation process. Using BIIF, each system must be compliant with only one external format that will be used for communication with all other participating systems. When BIIF is not used as a system's native internal format, each system will translate between the system's internal representation for imagery and the BIIF format. A system from which data is to be transferred has a translation module that accepts information structured according to the system's internal representation for images and related imagery data, and assembles this information into BIIF format. The receiving system will reformat BIIF data, converting it into one or more files structured as required by the internal representation of the receiving system. Each receiving system can translate selectively and permanently store only those portions of data in the received BIIF file that are of interest. A system may transmit all of its data, even though some of the receiving systems may be unable to process certain elements of the data. The functional architecture of this translation process is shown in Figure 1. In the diagram, the terms "Native Mode" refer to imagery and imagery related data represented in a way potentially unique to the sending or receiving system, respectively.

Flexibility and extensibility of the use of BIIF are provided through the use of a constrained set of conditional variable length fields and extension constructs. . Imagery, together with Programmer’s Imaging Kernel System (PIKS) objects, can easily be formatted into BIIF. This approach provides the proven capability to implement general purpose BIIF readers (applications) that can present the basic imagery and annotations of any BIIF compliant product file created within the constraints of a given profile of BIIF. Although more robust approaches exist to allow ‘self-defining’ data structures, these approaches significantly increase the complexity for implementing general purpose readers (applications) capable of meaningful interpretation of file constructs created by a wide variety of diversely developed generators. More simplistic imagery file formats also exist. They are often focused at just portraying a simple digital image and are often too limited in feature sets to meet the needs of somewhat more sophisticated, but still basic imagery applications. BIIF provides a basic capability that bridges the gap between simplistic digital image formats and the extremely sophisticated, self-defining, but potentially complex formats. As such, BIIF has some inherent bounds and limitations, but remains very capable as a basic imagery format capable of satisfying a broad range of imagery applications.

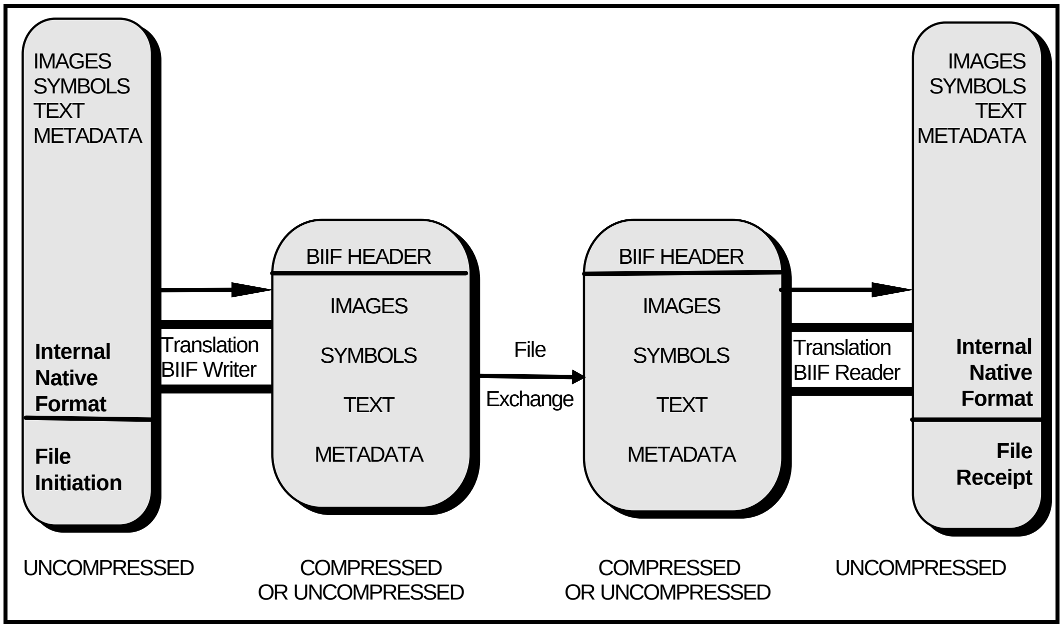


Figure 1 -- Translation process

BIIF supports extensibility (through the use of its Data Extension Segment (DES) and Reserved Extension Segment (RES)) which empowers new applications while maintaining backward compatibility. Newly defined data can be linked via a DES/RES and utilized by a new application or ignored by a legacy system. This will insure backward compatibility to older systems while incorporating new technology.

4.1.1 Description

4.1.1.1 Format fields

BIIF format structure consists of a combination of both fixed octet length fields and variable octet length fields. In its most basic application, the use of variable length fields is minimized to deal with the basic issues of the variable number and size of images and the variable number and size of image annotations that may appear in a BIIF file. The format contains header, subheader, and data fields. BIIF header and subheader fields are octet aligned. The file header carries information about the identification, security, structure, content, size of the file as a whole, and size of the data segments within the file. A data segment structure is defined for each kind of data supported by the format. Each data segment in the file has a subheader containing information that describes characteristics of the data segment and an associated data field that contains the actual data. BIIF encoding is discussed in Subclause 4.2.3.

4.1.1.2 Standard data types

A BIIF file supports inclusion of three standard types of data in a single file: image, symbol, and text. It is possible to include zero, one, or multiple data segments of each standard data type in a single file (for example: several images, but no symbols). Standard data types shall be placed in the file in the following order: all image data segments (images), followed by all symbol data segments (symbols including symbol text), followed by all text segments (documents or text). Additional kinds of data may be included in a BIIF file by use of Data Extension Segments (DES) (see Subclause 4.2.8), such as the Transportable File Structure (TFS Annex A), and Reserved Extension Segments (RES) (see Subclause 4.2.9). A data segment of a standard data type is called a standard data segment. A data segment of a type defined in a DES or RES is called an extension data segment. The order of these major file components is illustrated in Figure 2.

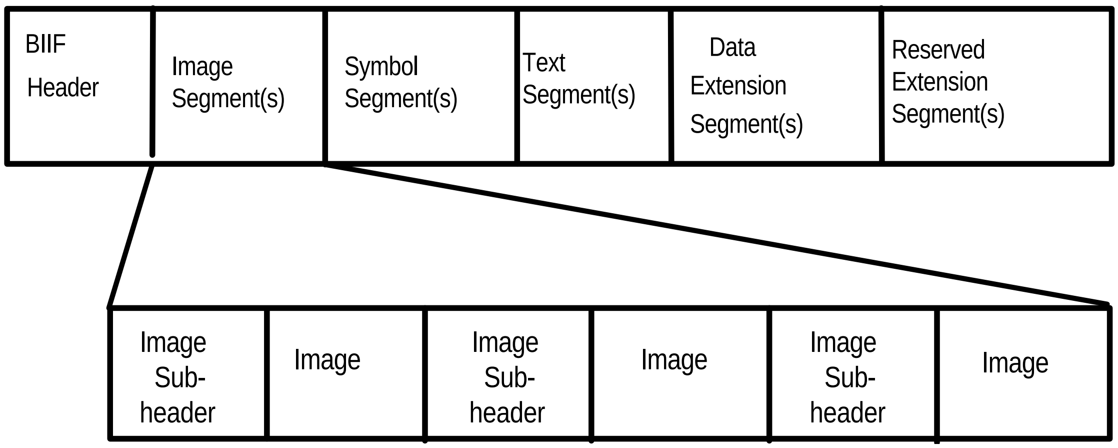


Figure 2 -- Structure

4.1.1.3 Extensions

Flexibility to add support for kinds of data and data characteristics not explicitly defined in this standard is provided within the format. This is accomplished by providing for one or two fields in each header/subheader containing "tagged record extensions" and by use of DESs and RESs. The tagged records extensions (TREs) in the header/subheaders may contain additional characteristics about the corresponding data segment, while the extension segments are intended primarily to provide a vehicle for adding support for new kinds of data. The identifier (tag name) for the tagged records, and extension segment identifiers, will be coordinated centrally in accordance with this standard to avoid conflicting use. In some cases, tagged record and extension formats will be configuration managed to control changes to data formats affecting a broad BIIF user base.

All implementations of BIIF should handle the receipt of unknown extension types by at least recognizing they are unknown extension types and ignoring them. This is accomplished through the octet count mechanism of the extension identifier plus length field approach for bounding separate content types in the format. The octet length offsets allow an interpret implementation to skip past unknown extensions and interpret the known meaningful content elements of the BIIF file. This concept is basic to BIIF interpretation; a reader can always identify octet count offsets and move on to other elements of the file. A quality implementation will have the provision to alert the user of unknown entities that are skipped over to ensure the user does not infer from the presentation of the file that all file contents have been fully interpreted and made available for use by the user.

The BIIF file header and each standard data segment subheader have designated expandable fields to allow for the optional inclusion of extension data (Tagged Record Extensions). The inclusion of extension data provides the ability to add data/information about the standard data segment (metadata) that is not contained in the basic fields of the headers and subheaders. The additional data is contained within one or more BIIF Tagged Record Extensions that are placed in the appropriate field (user defined data field or extended data field) of the standard data segment subheader for which the metadata applies. When Tagged Record Extensions have application across multiple data segments in the file, or otherwise apply to the entire BIIF file in general, they are placed in the appropriate file header extension fields.

Exemplary use of Tagged Record Extensions:

- Data about people, buildings, places, landmarks, equipment or other objects that may appear in the image.
- Data to allow correlation of information among multiple images and annotations within a BIIF file.
- Data about the equipment settings used to obtain the digital image, xray, etc
- Data to allow geopositioning of items in the imagery or measurement of distances of items in the imagery.

4.1.1.4 Transportable file structure (TFS)

The Transportable File Structure (TFS) Data Extension Element allows for the transport of the Programmer's Imaging Kernel System (PIKS) object data and other image related data. The TFS is a simple metafile that allows for both relational and hierarchical structures of image related data to be stored in BIIF. Each TFS contains one or more Transports. The Transports allow for the grouping of image related data for a specific reason. For example, the Transport could contain all the required information about a medical patient to include imagery, PIKS objects for the imagery, and a complete medical history of the patient. In the case of remote sensing, the Transport could contain area of interests, specific locations, and explanations of features on the image. Each Transport contains one or more Profiles which contain PIKS object data or other image related object data. TFS object data either contains data for the object or provides an unambiguous reference to the data. Profiles are related together when they occur at the same level in the metafile. For example, when a Transport contains a image Profile and several PIKS object Profiles, it is implied that the PIKS objects are related to the image. A Profile can also be nested within a Profile to represent complex hierarchical structures. Each TFS, Transport, and Profile contain metadata about itself and its contents, security about the data, and an index into the next levels for quick access. Although the TFS can be used to describe complex data structures about the image, PIKS data objects relating to an image can be simply expressed using one Transport and one Profile. The Transportable File Structure is completely described in Annex A. Example uses of TFS are given in Annex E.

4.1.1.5 Complexity Levels

The BIIF uses the concept of complexity levels to allow definitions of a set of nested features and constraints to exist within a user domain served by a specific profile of BIIF. It allows a user application to quickly identify the degree of complexity of the number and combinations of BIIF features used in any specific BIIF file.

4.1.2 Interoperability/exchange

Within a BIIF Profile, the term interoperability is used to express the ability of two or more imagery users within the same community of interest operating in a heterogeneous computing environment, to accurately create and/or recognize BIIF file structure as prescribed in this standard, and meaningfully exchange the information contained in it. This standard also promotes a higher degree of interoperability among two or more diverse communities of interest through the selection of a common set of functionality captured in specific profiles and implementation agreements.

4.1.3 Fields

4.1.3.1 Valid data

All header and subheader fields contained in the BIIF file shall contain valid data (that is, data in accordance with the restrictions specified for the contents of the field in this document or as constrained by an applicable profile definition; Annex C) or the specified default value. In this standard the word ‘profile’ shall refer to a registered BIIF profile that describes the field entries for each subclause of BIIF.

4.1.3.2 Date and time expressions

The representation of date and time shall be of the form CCYYMMDDhhmmss in accordance with the provisions of ISO 8601:1988(E) for expressing combined date and time of day representations with the century designator. All dates and times shall be expressed in Coordinated Universal Time (UTC).

The associated meaning of the CCYYMMDDhhmmss representation is:

- [CC] represents the digits used in the thousands and hundreds components of the time element “year”. The range of ‘CC’ is ‘00’ through ‘99’.
- [YY] represents the digits used in the tens and units components of the time element "year". The range of 'YY' is '00' through '99'.
- [MM] represents the digits used in the time element "month". The range of 'MM' is '01' through '12'.
- [DD] represents the digits used in the time element "day". The range of 'DD' is '01' through '31'.
- [hh] represents the digits used in the time element "hour". The range of 'hh' is '00' through ,'23'.
- [mm] represents the digits used in the time element "minute". The range of 'mm' is '00' through '59'.
- [ss] represents the digits used in the time element "second". The range of 'ss' is '00' through '59'.

4.1.3.3 Representation of textual information in fields

BIIF uses two different categories (BCS, UCS) of textual data character representations. Each category has a set of constraints for use within header and subheader fields. UTF 8 is used in selected header fields. BCS (BCS-A and BCS-N) and UCS (UTF 8) character representation are allowed in header and sub-header fields. BCS and UCS character codes are allowed for use in the data fields of text segments. (see following Clauses 4.1.3.3.1-4.1.3.3.5)

4.1.3.3.1 Basic character set

The Basic Character Set (BCS) is used when populating header and subheader fields of BIIF. These fields are primarily meant for internal use by computer systems and represent nothing more than an alphabetic code. Therefore the allowable characters are restricted to a relatively small set that can be represented in 8-bit per character codes. This character set is selected from ISO/IEC 646. In addition to header fields, BCS characters can also be used for text data segments. Valid BCS character codes range from 20 through FF and line feed (0A), formfeed (0B), and carriage return (0C).

4.1.3.3.2 Basic character set-numeric (BCS-N)

The range of allowable characters for BCS-N consists of the numbers '0' through '9' from the BMP block named 'BASIC LATIN', codes 30 through 39; plus sign, code 2B; minus sign, code 2D; decimal point, code 2E; and slash (/) , code 2F.

4.1.3.3.3 Basic character set-alphanumeric (BCS-A)

The range of allowable characters for BCS-A consists of the following: Space through Tilde, codes 20 through 7E (BMP block 'BASIC LATIN').

4.1.3.3.4 Universal multiple-octet coded character set (UCS)

The UCS is used for expressing text in many languages of the world as defined by ISO 10646. The specific character set selected from UCS shall be identified by profile. The profile shall identify the adopted form, the adopted implementation level and the adopted subset (list of collections and/or characters) in accordance with the structures defined in ISO 10646. When a profile defined UCS is used in a BIIF file, the coding shall contain an explicit declaration of identification of features (escape sequence) as specified in ISO 10646. When no declaration escape sequence is included, the default shall be that defined for BCS above. UCS-2 and UCS-4 are respectively, the two and four octet forms.

4.1.3.3.5 UCS Transformation Format 8 (UTF-8)

UTF-8 is the UCS Transformation Format 8 defined in ISO 10646 Part 1, Amendment 2 that allows the use of multi-national character sets in BIIF. It is an alternative coded representation for all of the characters of the UCS. UCS characters from the BASIC Latin collection are represented in UTF-8 with single octet values ranging from 20 to 7E. Octet values 00 to 7F do not otherwise occur in the UTF-8 coded representation of any character. Therefore, the BASIC Latin collection is unchanged under a UTF-8 transformation. In UTF-8 each character has a coded representation that comprises a sequence of octets of length 1, 2, 3, 4, 5, or 6 octets. For all sequences of one octet, the most significant bit shall be a ZERO bit. All octets other than the first in a sequence are continuing octets. The number of ONE bits in the most significant positions of the first octet of the sequence determines how many continuing octets represent the international character. Selected header and subheader fields allow UTF-8 coded characters. Field lengths are fixed octet counts regardless of character encoding.

4.1.4 Logical structure of pixel storage

An image band is stored in row, column order, such that the column index is changing faster than the row index. The band index may be changing either faster than the column index or slower than the row index. The orientation of the pixels in the image array to up, down, left and right is defined by the metadata associated with the image. The image array structure is depicted in Figure 3. The origin of the image array I, pixel I(0,0), is at the upper left corner, and pixel I(R-1,C-1) is at the lower right corner.

Within the r^{th} row, the pixels shall appear beginning on the left with I(r,0) and proceeding from left to right with I(r,1), I(r,2) and so on, ending with I(r,C-1).

4.1.4.1 Pixel coordinates

The black square in Figure 3 is a single pixel located by (r,c) where r=row number, and c=column number. Each pixel value for each band in an image can be uniquely identified by its (r, c) coordinate.

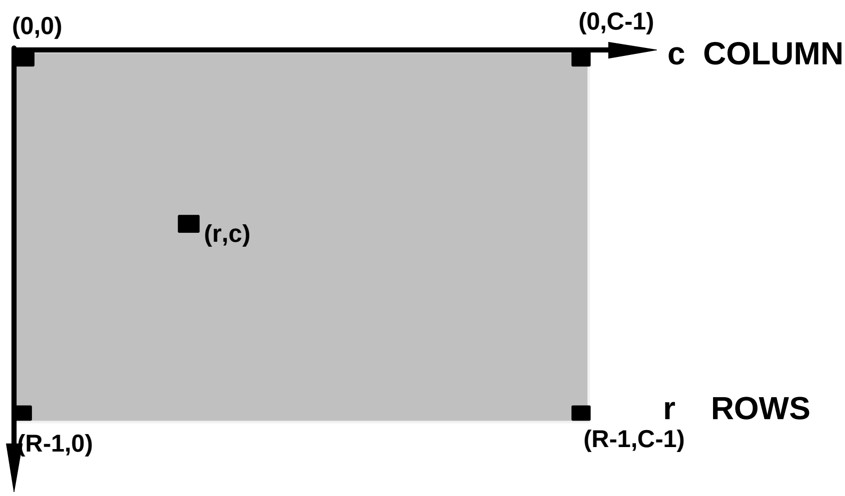


Figure 3 -- Image Array Structure

4.1.4.2 PIKS pixel coordinates

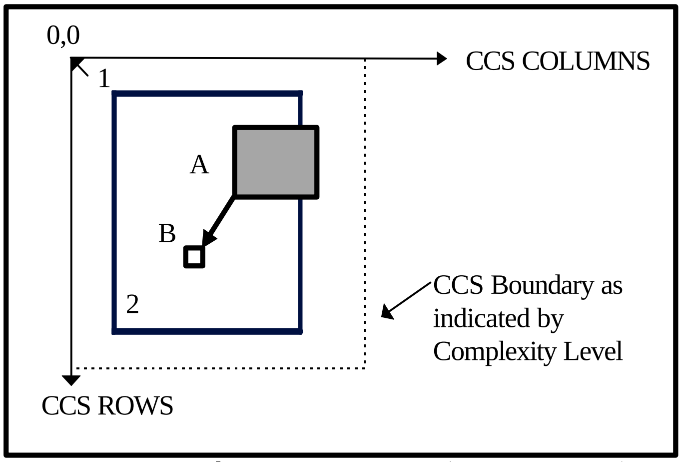
The Image Array pixel coordinate indexing (row, column) demonstrated in Figure 3 is typical for image scanning systems (e.g. line, sample), but is reversed from that used by PIKS. In PIKS, the first index is across columns. Implementors must take this into consideration when passing pixel coordinates through the PIKS Application Programming Interface (API). The pixel storage order is the same, only the coordinate indexing convention is reversed.

4.1.5 Common coordinate system

The BIIF Common Coordinate System (CCS) is the two dimensional coordinate space which shall be common for determining the placement and orientation of displayable data types (e.g. images, symbols, extension data, etc.) within a specific BIIF file and among correlated BIIF files which comprise an integrated product.

4.1.5.1 Common coordinate system structure

The Common Coordinate System structure can be conceived of as a virtual two dimensional drawing space with a coordinate system similar in structure to the lower right quadrant of the Cartesian coordinate system. The CCS has two perpendicular coordinate axes, the horizontal column axis and the vertical row axis as depicted in Figure 4. The positive directions of the axes are based on the predominate scan (column) and line (row) directions used by the digital imagery community. The intersection of the axes is designated as the origin point with the coordinates (0,0). Given the orientation of the axes in Figure 4, the positive direction for the column axis is from (0,0) and to the right; the positive direction for the row axis is from (0,0) downward. The quadrant represented by the positive column and positive row axes is the only coordinate space for which BIIF displayable data types may be located.



1. Common Coordinate System Origin (0,0 - 2048,2040)
2. Image
A. Image
B. Graphic (Box & Arrow)

Figure 4 -- Common coordinate system

4.1.5.2 Row and column coordinates

Images and symbols have designated points of references. For an image it is the first pixel (0,0). For a CGM symbol, it is the CGM virtual display coordinates (VDC) origin point. Symbols and images shall be placed in the CCS according to the CCS row and column coordinates placed in subheader location fields (e.g., ILOC, SLOC). The location coordinates of a specific image or symbol represent row and column offsets from either the CCS origin point when ‘unattached’, or the location point in the CCS of the data to which it is attached. Other means used to locate displayable data shall be directly correlated to row and column coordinates (e.g., displayable tagged record extension data might have geolocation data correlated with row and column indices). Note: When passing coordinate pairs to a PIKS API, the order of the coordinates need to be reversed by the application.

When location coordinates are relative to the CCS origin, they shall always have a positive value. When location coordinates are relative to the location coordinates of an item to which they are attached, both positive and negative offset values are possible. In all cases, the location coordinates selected for any data item shall ensure that none of the displayable item extends outside of the quadrant defined by the axes of the CCS.

4.1.5.3 Complexity level constraints

The upper and left boundaries of the CCS are explicitly constrained in the specification. The lower and right boundaries constraints are defined as one of the key attributes of the complexity level definition in a profile. Image and symbol segments shall be displayable and located such that no displayable portion of any image or subimage in a BIIF file extends beyond the CCS boundaries applicable to the file according to the selected complexity level.

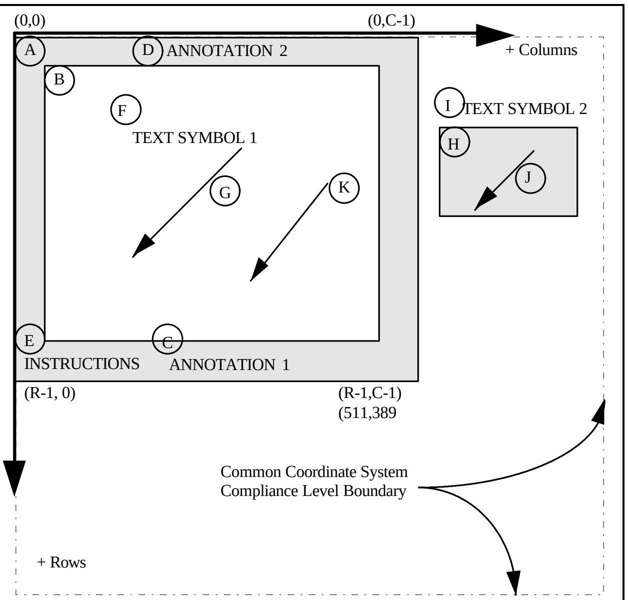
4.1.6 Display and attachment levels

Each image product shall be comprised of one or more data segments. The relative visibility of the various items in BIIF is recorded by use of the display level (the "DLVL" field in the standard data segment subheaders, specifically IDLVL for image, SDLVL for symbol). Groups of related items may be formed by use of the attachment level (the "ALVL" field in the standard data segment subheaders, specifically IALVL for image, SALVL for symbol).

4.1.6.1 Display levels

The order in which images and symbols are "stacked" shall be determined by their display level, not by their relative position within a BIIF file. Every image and symbol component in a BIIF file shall have a unique display level. This requirement allows "stacking" to be independent of data sequence or processing order.

Figure 5 illustrates a sample "output presentation" from a BIIF file that illustrates the effects of display level assignment. The display level of each segment shown on Figure 5 is indicated in the list of segments, where the list is in the order that the segments were placed in BIIF containing them. In the case shown, the segment with display level one is not an image but rather an opaque CGM rectangle (symbol data, not image data). Because the CGM rectangle is larger than the image (which, in this case, serves as the first overlay because its display level is two), it provides a border to the image. Following increasing DL value, the border is overlaid by the image being examined which, in turn, is overlaid by arrow one, which is in turn overlaid by the image inset, which is overlaid by the annotation, which is overlaid by the arrow annotation, etc. The AL values in Figure 4 refer to "Attachment Levels".



	Location Offsets			
	DL	AL	Row	Column
A. Border (Opaque Box)	001	000	00000	00000
B. Exploited Image	002	001	00025	00025
C. Annotation 1	999	000	00370	00256
D. Annotation 2	998	000	00015	00256
E. Instructions	997	000	00370	00040
F. Text Symbol 1	003	002	00060	00240
G. Arrow	004	003	00010	00040
H. Image Inset	005	002	00045	00525
I. Text Symbol 2	006	005	00020	00050
J. Arrow 2	007	006	00010	00050
K. Arrow 3	008	005	00070	-0005

Figure 5 -- Display level and attachment level relationships

4.1.6.2 Attachment levels

The attachment level (AL) provides a way to associate items (images, symbols) so they may be treated together for certain operations such as moving, rotating, or displaying. The attachment level of an item shall be equal to the display level of the item to which it is "attached." This value is stored in the "ALVL" field (specifically IALVL for images, SALVL for symbols) of the item's subheader. The segment with the lowest display level (display level 001 in the example in Figure 5), must have an attachment level of zero. An attachment level of zero shall be interpreted as "unattached." Any other item may also have AL 000, that is, be unattached.

Figure 6 shows the attachment relationships of overlay items in Figure 5. When an overlay or base is edited (moved, deleted, rotated), all overlays attached to it, directly or indirectly, shall be affected by the same operation. For example, in Figure 6, if the inset being examined (DL 005, AL 004) was moved one centimeter to the left, the arrows (DL 008, AL 005, and DL 007, AL 006), and symbol (DL 006, AL 005) associated with the examined inset (DL 005, AL 004) would also be moved one centimeter to the left. Also note that because of the way the attachments have been constructed, if the symbol (DL 005, AL 004) were deleted, so would arrow 2 (006, 005). However, if the image inset (DL 006, AL 005) were deleted, arrow (DL 004, AL 005) would not be deleted.

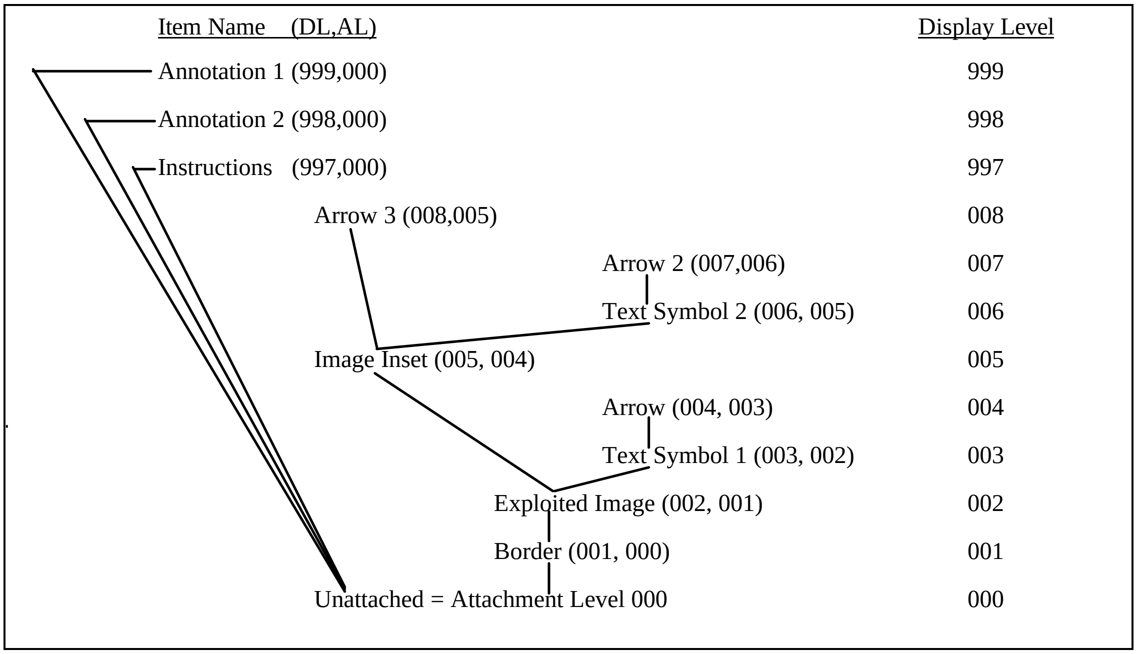


Figure 6 -- Display and attachment levels

4.2 Format

The format of the BIIF file consists of a header, followed by data type segments with their associated subheaders. The header specifies profile and structural information that allows proper interpretation of the rest of the header and subheaders. All headers and subheaders have their character data specified in the lexical constraints of BCS-A, BCS-N or UTF-8. Data in the text segments may be specified in other lexical levels of ISO 10646 as discussed in Subclause 4.1.3. The header is fixed in structure to support easy location and interpretation of the data. This is especially important for security-related information since this insures that security data is immediately accessible. Additionally, a fixed file structure allows an additional level of data integrity checking since octet counts are known in advance.

Each image, symbol, text, and extension data type included in BIIF shall be preceded by a "subheader" corresponding to that data item. This comprises a data segment. This subheader shall contain information pertaining to that particular data item and data type only. If no segments of a given type are included in BIIF, a subheader for that data type shall not be included in BIIF. A BIIF file may contain multiple segments of each data type (image, symbol, text, and extensions). All segments of a specific data type shall precede the segments of the next data type.

All data items and the associated subheader of a single type shall precede the first subheader for the next data type. Field values are to be read as a continuous stream as defined in the tables below.

4.2.1 Data recording formats.

The method of converting data into an octet string shall adhere to the following:

1) octet oriented data (BCS character string) shall be recorded with no change

2) the ordering sequence for bit oriented data (integers, reals) shall be recorded in the order where the left-most eight bits of remaining data are always output first. See Figure 7 where octet 0 is the most-significant (leftmost) octet within a multi-octet data structure.

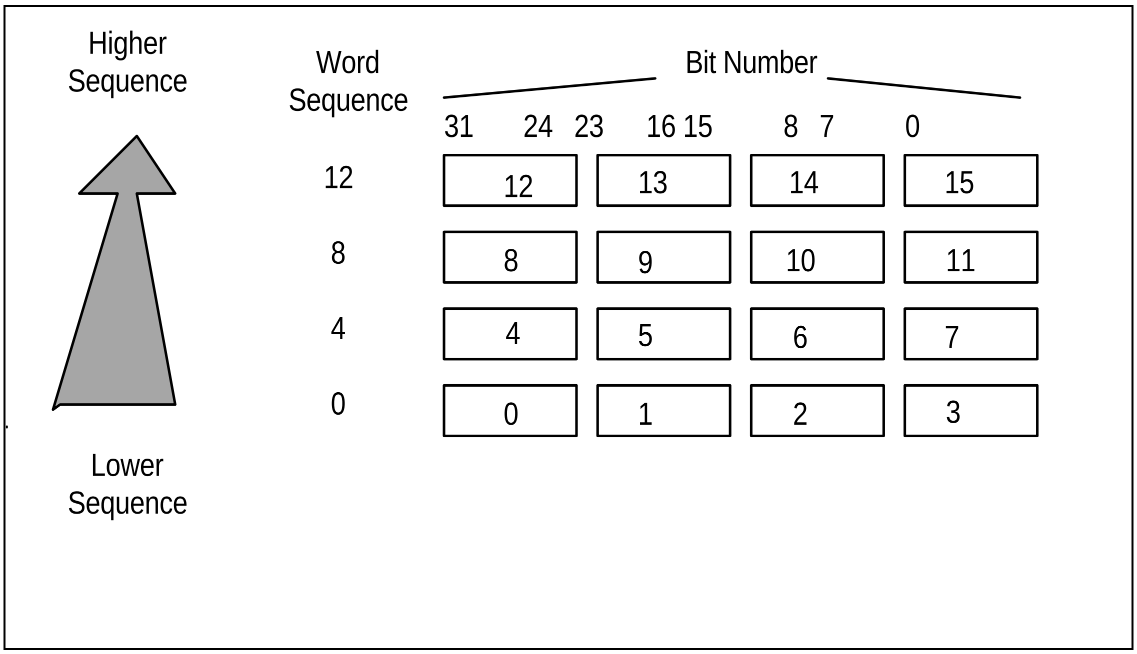


Figure 7 -- Octet sequence order

4.2.2 Encoding

The tables in the following subclauses specify the encoding structure of a BIIF file. BIIF specifies a single encoding consisting of both a fixed and variable field sizes and a fixed field ordering. Compound elements (headers, subheaders, etc.) are constructed by packing primitive elements (fields, attributes) in a fixed order including or excluding certain condition dictates. The tables below provide the syntax/format for each of the primitive elements. A separate table is provided for each compound element. The following notation is used in the tabular entries of the five column tables which follow:

Column I. Field name: A short name used for references in the text descriptions.
Column II. Description: A short description of the meaning of the field, followed by a more detailed explanation as required.
Column III. Type A selection from the following codes:
R: Required element; structure (syntax) and meaning of this element shall be implemented precisely as specified in this standard. It must be present.
C: Conditional element; this element is omitted based on the value of its “dependent element”. For example, if NUMI (number of images) = 0, then no image components (image subheader, image data) are present in the file. A conditional field may or may not be present depending on the value of one or more preceding required fields.
PVV: Profile Variant-Value: A restriction on value range for this element is permitted in a profile. For example, a profile might be developed that supported only one image in the file. In this case, NUMI (number of images) would be constrained to the value one for all BIIF files

conforming to this profile. The profile is required to identify constraints on the data value range.

PVU: Profile Variant - Unspecified: Both structure (syntax) and meaning are allowed to be defined by the profile; however, size and order within compound element must be maintained in accordance with this standard. The PVU and PVV designation may be applied to either an R or C field and will be shown, for example, as R/PVV or C/PVU.

Column IV. CE/Size: Character Encoding: A = BCS-A; N =BCS-N, U8=UTF-8. Size: Equals the number of octets that are reserved for the field. Size is fixed and must be filled with valid data or the specified default.
For those “Required” fields BCS-A “spaces” will be applied as default, for fields labeled BCS-N “leading zeros (0)” will be applied, for fields labeled U8, “spaces” will be applied. E.g., U8/80 - UTF8 encoded with a total length of 80 Octets, default is 80 “spaces”.

Column V Value Range: Valid information must fall within the ranges identified and may be a range, an enumerated set, or a single value.

The specification of the fields in the various subheaders found within the BIIF file is provided in a series of tables. For each field in the subheader, the table includes a mnemonic field identifier, the field's name and description of the valid contents of the field, the field BCS type and size, and the range of values it may contain, and any constraints on the field's use. The Tagged Record Extension headers are defined in Table 7. The data that appears in all subheader information fields specified in the tables, including numbers, shall be represented using the basic character set with eight bits (one octet) per character. All field size specifications given for the header and subheader fields specify a number of octets.

4.2.3 Header

Each BIIF file shall begin with a header whose fields contain identification, origination information, security information, and the number and size of data items of each type contained in BIIF. Figure 8 depicts the header. It illustrates the types of information contained in the header and shows the headers organization as a sequence of groups of related fields. The expansion of the "Image Group" illustrates how the header's overall length and content may expand or contract depending on the number of data items of each type included in BIIF.

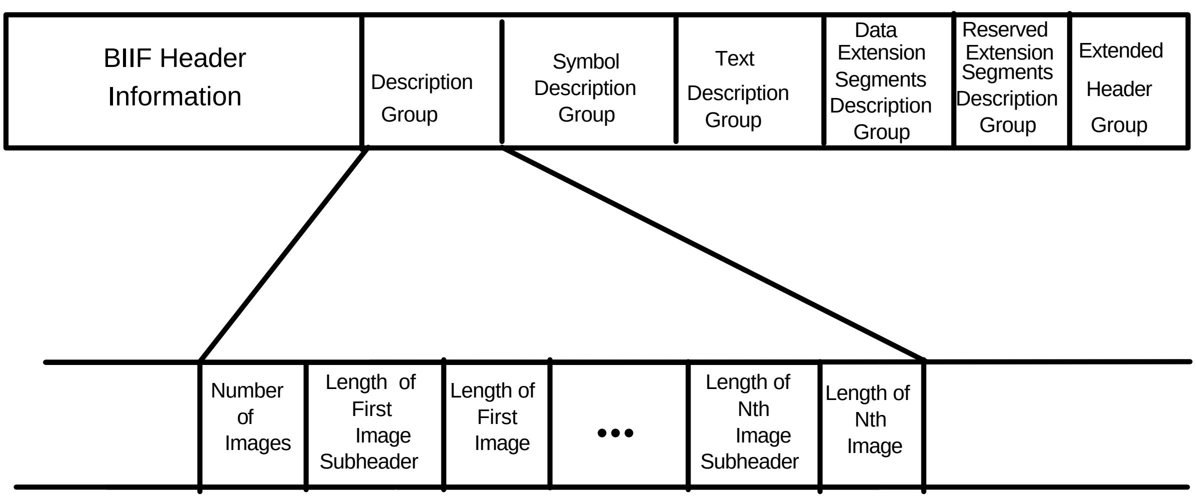


Figure 8 -- Header structure

Table 1 -- Header

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
FHDR	Profile Name A character string uniquely identified in the profile.	R/PVV	A/4	Only approved and registered profile short names
FVER	Version A character string uniquely denoting the version.	R/PVV	A/5	Only approved and registered versions
CLEVEL	Profile Complexity level A character string defined in the profile for BIIF complexity level (such as 00 through 99).	R/PVV	A/2	Only approved and registered profile complexity levels or 00 for non-hierarchical profiles
STYPE	Standard Type Identifies the file as being formatted according to the BIIF standard.	R	A/4	BF01
OSTAID	Originator System ID This field shall contain an identification indicator of the originating system.	R/PVU	A/10	As specified in profile.
FDT	File Date & Time This field shall contain the date and time of the BIIF file origination. The time shall be specified in UTC (Z). A software implementation may display the time in any desired format; however, data recorded in a BIIF file shall be recorded in UTC.	R	N/14	CCYYMMDDhhmmss
FTITLE	File Title This field shall contain the title of the BIIF file.	R/PVU	U8/80	As specified in profile.
FSEC	File Security Profile Specific Parameters This field shall contain profile specific information for the image product security as defined in the profile.	R/PVU	U8/167	As specified in profile.
FSCOP	File Copy Number This field shall contain the copy number of the file.	R/PVV	N/5	00000 default, or actual number.
FSCPYS	File Number of Copies. This field shall contain the total number of copies of the file.	R/PVV	N/5	00000 default or actual count.
ENCRYP	Encryption Encryption codes and meaning as defined by profile.	R/PVU	A/1	0 = not encrypted, other codes as specified by profile.
OID	Originator's ID or other file information.	R/PVU	A/45	As specified in profile.

Table 1 -- Header *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
FL	File Length This field shall contain the length in octets of the entire BIIF file including all headers, subheaders, and data.	R/PVV	N/12	"000000000388"- "999999999999"
HL	Header Length This field shall contain a valid length in octets of the BIIF header.	R	N/6	"000388"- "999999"
NUMI	Number of Images This field shall contain the number of separate images included in the BIIF file. This field shall be zero if and only if no images are included in the BIIF file.	R/PVV	N/3	"000"- "999"
LISH001	Length of 1st Image Subheader If the field NUMI contains a value of one or more, this field shall contain a valid length in octets for the subheader of the first image in the BIIF file. This field is conditional and shall be omitted if NUMI field contains zero.	C	N/6	"000439"- "999999"
LI001	Length of 1st Image This field shall contain a valid length in octets of the first image. This field is conditional and shall be omitted if NUMI field contains zero.	C	N/10	"0000000001"- "9999999999"
.....				
LISHn	Length of nth Image Subheader This field shall contain a valid length in octets for the nth image subheader, where n is the number of the image counting from the first image in order of the images' appearance in the BIIF file. This field shall occur as many times as specified in the NUMI field. This field is conditional and shall be omitted if NUMI field contains a value less than n. Possible values of n are 001 to 999.	C	N/6	"000439"- "999999"

Table 1 -- Header *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
LIn	Length of nth Image This field shall contain a valid length in octets of the nth image, where n is the image number of the image counting from the first image in order of the images' appearance in the BIIF file. If the image is compressed, the length after compression shall be used. This field shall occur as many times as specified in the NUMI field. This field is conditional and shall be omitted if NUMI field contains a value less then n. Possible values of n are 001 to 999	C	N/10	"0000000001"- "9999999999"
NUMS	Number of Symbols This field shall contain the number of separate symbols included in the BIIF file. This field shall be zero if and only if no symbols are included in the BIIF file.	R/PVV	N/3	“000"-"999"
LSSH001	Length of 1st Symbol Subheader If the field NUMS contains a value of one or more, this field shall contain a valid length in octets for the subheader of the first symbol in the BIIF file. This field is conditional and shall be omitted if NUMS field contains zero.	C	N/4	"0258"-"9999"
LS001	Length of 1st Symbol This field shall contain a valid length in octets for the first symbol. This field is conditional and shall be omitted if NUMS field contains zero.	C	N/6	"000001"- "999999"
.....				
LSSHn	Length of nth Symbol Subheader This field shall contain a valid length in octets for the nth symbol subheader, where n is the number of the symbols counting from the first symbol in the order of the symbol’s appearance in the BIIF file. This field shall occur as many times as specified in the NUMS field. This field is conditional and shall be omitted if NUMS contains zero. Possible values of n are 001 to 999.	C	N/4	"0258"-"9999"

Table 1 -- Header *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
LSn	Length of n^{th} Symbol This field shall contain a valid length in octets of the n^{th} symbol, where nnn is the symbol number of the symbol, counting from the first symbol in the order of the symbol's appearance in the BIIF file. This field shall occur as many times as specified in the NUMS field. This field is conditional and shall be omitted if NUMS field contains a value less then n. Possible values of n are 001 to 999.	C	N/6	"000001"- "999999"
NUMX	Reserved for Future Segment Types	R	N/3	"000"
NUMT	Number of Text Segments This field shall contain the number of separate text segments included in the BIIF file. The value is valid only if it is within the specified range. This field shall be zero if and only if no text items are included in the BIIF file.	R/PVV	N/3	"000"-"999"
LTSH001	Length of 1^{st} Text Subheader If the field NUMT contains a value of one or more, this field shall contain a valid length in octets for the subheader of the first text item in the BIIF file. This field is conditional and shall be omitted if NUMT field contains zero.	C	N/4	"0282"-"9999"
LT001	Length of 1^{st} Text Item This field shall contain a valid length in octets for the first text item. This field is conditional and shall be omitted if NUMT field contains zero.	C	N/5	"00001"-"99999"
.....				
LTSHn	Length of n^{th} Text Subheader This field shall contain a valid length in octets for the n^{th} text segment subheader, where n is the number of the text segments, counting from the first text segment in the order of the text segment's appearance in the BIIF file. This field shall occur as many times as specified in the NUMT field. This field is conditional and shall be omitted if NUMT field contains a value less then n. Possible values of n are 001 to 999.	C	N/4	"0282"-"9999"

Table 1 -- Header *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
LTn	Length of n^{th} Text item This field shall contain a valid length in octets of the n^{th} text item, where n is the number of the text segment, counting from the first text segment in the order of the text segment's appearance in the BIIF file. This field shall occur as many times as specified in the NUMT field. This field is conditional and shall be omitted if NUMT field contains less than n. Possible values of n are 001 to 999.	C	N/5	"00001"-"99999"
NUMDES	Number of Data Extension Segments This field shall contain the number of separate data extension segments included in the BIIF file. This field shall be zero if and only if no data extension segments are included in the BIIF file.	R/PVV	N/3	"000"-"999"
LDSH001	Length of 1^{st} Data Extension Segment Subheader If the field NUMDES contains a value of one or more, this field shall contain a valid length in octets for the subheader of the first data extension segment in the BIIF file. This field is conditional and shall be omitted if the NUMDES field contains zero.	C	N/4	"0200"-"9999"
LD001	Length of 1^{st} Data Extension Segment Data Field This field shall contain a valid length in octets for the data field of the first data extension segment. This field is conditional and shall be omitted if the NUMDES field contains zero.	C	N/9	"000000001"- "999999999"
.....				
LDSHn	Length of n^{th} Data Extension Segment Subheader This field shall contain a valid length in octets for the n^{th} extension segment subheader, where n is the number of the data extension segment counting from the first data extension segment in order of the data extension segment's appearance in the BIIF file. This field shall occur as many times as are specified in the NUMDES field. This field is conditional and shall be omitted if the NUMDES field contains a value less than n. Possible values of n are 001 to 999.	C	N/4	"0200"- "9999"

Table 1 -- Header *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
LDn	Length of n^{th} Data Extension Segment Data Field This field shall contain a valid length in octets of the data field of the n^{th} data extension segment, where n is the number of the data extension segment counting from the first data extension segment in order of the data extension segment's appearance in the BIIF file. This field shall occur as many times as are specified in the NUMDES field. This field is conditional and shall be omitted if the NUMDES field contains a value less than n. Possible values for n are 001 to 999.	C	N/9	"000000001"- "999999999"
NUMRES	Number of Reserved Extension Segments This field shall contain the number of separate reserved extension segments included in the BIIF file. This field shall be zero if and only if no reserved extension segments are included in the BIIF file.	R/PVV	N/3	"000"-"999"
LRSH001	Length of 1^{st} Reserved Extension Segment Subheader If the field NUMRES contains a value of one or more, this field shall contain a valid length in octets for the subheader of the first reserved extension segment in the BIIF file. This field is conditional and shall be omitted if the NUMRES field contains zero.	C	N/4	"0000"-"9999"
LR001	Length of 1^{st} Reserved Extension Segment Data Field This field shall contain a valid length in octets for the data field of the first reserved extension segment. This field is conditional and shall be omitted if the NUMRES field contains zero.	C	N/7	"0000001"- "9999999"
.....				

Table 1 -- Header *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
LRSHn	Length of n^{th} Reserved Extension Segment Subheader This field shall contain a valid length in octets for the n^{th} reserved segment subheader, where n is the number of the reserved extension segment counting from the first reserved extension segment in order of the reserved extension segment's appearance in the BIIF file. This field shall occur as many times as are specified in the NUMRES field. This field is conditional and shall be omitted if the NUMRES field contains a value less than n. Possible values for n are 001 to 999.	C	N/4	"0000"-"9999"
LRn	Length of n^{th} Reserved Extension Segment Data Field This field shall contain a valid length in octets of the data field of the n^{th} reserved extension segment, where n is the number of the reserved extension segment counting from the first reserved extension segment in order of the reserved extension segment's appearance in the BIIF file. This field shall occur as many times as are specified in the NUMRES field. This field is conditional and shall be omitted if the NUMRES field contains a value less than n. Possible values for n are 001 to 999.	C	N/7	"0000001"- "9999999"
UDHDL	User Defined Header Data Length This field shall contain the length in octets of the entire UDHD field plus three (3) octets. The length is three (3) plus the sum of the lengths of all the Tagged Record Extensions (TREs) appearing in the UDHD field, since they are not separated from one another. A value of zero shall mean that no Tagged Record Extensions are included in the UDHD field. If a Tagged Record Extension is too long to fit in the UDHD field, it may be put in a Data Extension Segment (DES). See Section 4.2.8 on Tagged Record Extensions.	R/PVV	N/5	"00000"-"99999"
UDHOFL	User Defined Header Overflow If present, this field shall contain "000" if the Tagged Record Extensions in UDHD do not overflow into a DES, or shall contain the sequence number of the DES into which they do overflow. This field shall be omitted if the field UDHDL contains zero.	C	N/3	"000-999"

Table 1 -- Header *(concluded)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
UDHD	User Defined Header Data If present, this field shall contain Tagged Record Extensions (TREs) as allowed by the profile. The length of this field shall be the length specified by the UDHDL field minus three (3) octets. The TREs shall appear one after the other with no intervening octets. This field shall be omitted if the field UDHDL contains zero.	C/PVU	TREs, Length specified by UDHDL minus 3	TREs as allowed by Profile
XHDL	Extended Header Data Length This field shall contain the length in octets of the entire XHD field plus three (3). The length is three (3) plus the sum of the lengths of all the Tagged Record Extensions appearing in the XHD field, since they are not separated from one another. A value of zero shall mean that no TREs are included in the XHD field. If an TRE is too long to fit in the XHD field, it may be put in a Data Extension Segment (DES).	R/PVV	N/5	"00000"-"99999"
XHDLOFL	Extended Header Data Overflow If present, this field shall contain "000" if the Tagged Record Extensions in XHD do not overflow into a DES, or shall contain the sequence number of the DES into which they overflow. The XHDLOFL field shall be omitted if the XHDL field contains a value of zero.	C	N/3	"000-999"
XHD	Extended Header Data If present, this field shall contain Tagged Record Extensions as allowed by the profile. The length of this field shall be the length specified by the field XHDL minus three octets. Tagged record extensions shall appear one after the other with no intervening octets. This field shall be omitted if the field XHDL contains zero.	C/PVU	TREs, Length specified by XHDL minus three (3)	TREs as allowed by Profile

4.2.4 Image segment

4.2.4.1 Image subheader

In a BIIF file, the information describing an image segment is represented in a series of adjacent fields grouped into the image subheader followed by the image data. The field containing the image data is called the image data field. The image data field shall follow immediately the last field of the corresponding image subheader with no intervening special characters to designate the beginning of the image data field. Similarly, the image subheader of the first image shall follow immediately the last octet of data of the last field in the BIIF file header. The image subheader of successive images shall follow immediately the last octet of the image data field of the preceding image. Valid data types, and their minimum levels of precision, are given in Table 2. The maximum level of precision for each data type is 96 bits, or as constrained by profile. This is a restriction of the field NBPP in the image subheader. The table indicates the correspondence between BIIF pixel data types, PIKS pixel data types, and 12087-1 data types.

BIIF provides for the transport of PIKS image objects and PIKS image-related non-image objects for applications in which the BIIF basic image and non-image data objects do not provide desired functionality. The transport mechanism for PIKS objects shall be the Transportable File Structure. The PIKS data objects are semantically specified in ISO/IEC 12087-1, Clause 5.4.

Table A.18 lists the size limits for PIKS parameters (BP, NP, SP, RP, CP, and CS) and pixel (BD, ND, SD, RD, CD) data types. In some cases, the minimum size limit for a PIKS pixel data type is greater than the minimum size limit for the corresponding BIIF pixel data type. If a BIIF image is imported into a PIKS application, it is possible that pixels may have to be padded to PIKS minimum lengths. Table A.18 lists the data structures of the PIKS image and non-image data objects derived from ISO/IEC 12087-1, Clause 5.4. These objects include:

PIKS_IMAGE
PIKS_HISTOGRAM
PIKS_LOOKUP_TABLE
PIKS_MATRIX
PIKS_NEIGHBOURHOOD
PIKS_PIXEL_RECORD
PIKS_STATIC_ARRAY
PIKS_TUPLE
PIKS_ROI_COORDINATE
PIKS_ROI_ELLIPTICAL
PIKS_ROI_POLYGON
PIKS_ROI_RECTANGLE
PIKS_VALUE_BOUNDS

PIKS images provide expanded capability for representation from the “Baseline” capabilities of BIIF (e.g., baseline BIIF has homogeneous pixel values for monochrome or colour images while PIKS images may have heterogeneous values of up to five dimensions). The PIKS dimensions are in the following slice order:

x-y Spatial slice
z Volumetric slices
t Temporal slices
b Colour or multispectral slices

The PIKS image object data shall be placed in the transport data stream immediately following the image subheader, as specified in Table 3. The PIKS non-image object data shall be placed in the transport data stream immediately following the data object attributes for each object, as specified in Table A.18.

Table 2 -- Image pixel data type codes and descriptions

PVTYPE	DESCRIPTION	Related PIKS Codes	12087-1 Data Types
B	1 bit Boolean values - On or off	BD	BC
INT	2 bit or greater unsigned integer	ND	NC
SI	2 bit or greater signed integer	SD	SC
R	Real arithmetic, 32-bit floating point representation	RD	RC
C	Complex arithmetic, values shall be represented with real and imaginary parts; each in 32-bit floating point representation and appearing in adjacent blocks, first real, then imaginary	CD	CC

4.2.4.2 Look-up tables (LUTS)

The BIIF provides a basic mechanism for associating simple look up tables in the image subheader for use in conjunction with the image data. Additional flexibility in defining and using look-up tables can be obtained through the use of PIKS object classes as defined in the TFS, Annex A.

4.2.4.2.1 Gray scale look up tables (LUT)

The gray scale to be used in displaying each pixel of a gray scale image is determined using the image’s LUT, if present. A LUT for a gray scale image shall comprise a one octet entry for each integer (the entry’s index) in the range 0 to NELUT-1. The octets of the LUT shall appear in the file one after the other without separation. The entries shall occur in the index order, the first entry corresponding to index 0, the second to index 1, and so on, the last corresponding to index NELUT-1. The display shade for a pixel in the image shall be determined by using the image pixel value as an index into the LUT. The LUT value shall correspond to the display gray scale in a way specific to the display device. NELUT shall be equal to or greater than the maximum pixel value in the image to ensure that all image pixels are mapped to the display device.

4.2.4.2.2 Colour look-up tables (LUT)

Colour images are represented using the RGB colour system notation. For colour images, each LUT entry, if present, shall be composed of the output colour components red, green, and blue, appearing in the file in that order. There shall be a LUT entry for each pixel value in a particular band of a BIIF image (the entry's index of the LUT will range from 0 to 2^{NBPP}-1). The LUT entries shall appear in the file in increasing index order beginning with index 0. The display colour of an image pixel shall be determined by using the pixel value as an index into each LUT (red, green, blue). The corresponding values for red, green, blue shall determine the displayed colour in a manner specific to the display device. The colour component values may be any of the 256 pixel values associated with the band. The presence of colour LUT's is optional for 24-bit per pixel (true colour) images. Pseudo-colour (e.g., 8-bit per pixel colour images) shall contain a LUT to correlate each pixel value with a designated true colour value.

Table 3 -- Image subheader

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
IM	BIIF Image Subheader This field shall contain the characters "IM" to identify the subheader as an image subheader.	R	A/ 2	"IM"
IID	Image ID This field shall contain an image identification.	R/PVU	A/10	As specified by profile.
IDATIM	Image Date & Time This field shall contain the time of the image origination.	R	N/14	CCYYMMDDhhmmss
IINFO	Image Information This field contains information about the main items of interest in the image.	R/PVU	U8/97	As specified by profile.
ISCSEC	Image Security Profile Specific Parameters This field shall contain profile specific information for the image segment security.	R/PVU	U8/167	As specified by profile.
ENCRYP	Encryption Encryption codes and meaning as defined by profile.	R/PVU	A/1	0 = not encrypted. Other codes as specified by profile.
ISORCE	Image Source This field contains the source of the image.	R/PVU	U8/42	As specified by profile.
NROWS	Number of Valid Rows in image This field contains the total number of rows of valid pixels in the image.	R/PVV	N/8	"00000001"- "99999999"
NCOLS	Number of Valid Columns in image This field contains the total number of columns of valid pixels in the image.	R/PVV	N/8	"00000001"- "99999999"
PVTYPE	Pixel data representation type This field shall contain an indicator of the type of computer representation used for the value for each pixel for each band in the image.	R/PVV	A/3	As specified in Table 2

Table 3 -- Image subheader *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
IREP	Image Representation This field shall contain a valid indicator for the general kind of image represented by the data.	R/PVV	A/8	MONO, RGB, RGB/LUT, HIS, CMY, CMYK, YIQ, YUV, YCbCr, CIE, 1D, 2D, ND, MULTI, PIKS - additional values specified through the graphical item registration process.
ICAT	Image Category This field shall contain a valid indicator of the specific category (often revealing the nature of the collector or intended use).	R/PVU	A/8	VIS - Visual imagery SL - Side looking radar TI - Thermal infrared FL - Forward looking IR RD - Radar EO - Electro-optical OP - Optical HR - High res. radar HS - Hyperspectral CP - Colour frame photo BP - Black/white photo SAR - Synthetic Aperature Radar SARIQ - SAR radio hologram IR - Infrared MS - Multi-spectral FP - Finger prints MRI - Magnetic Resonance imagery XRAY - x-rays CAT - CAT scan MAP - Image map PAT - Colour patch LEG - Legend DTEM - Elevation model data MATR - general matrix data LOCG - Location grids - additional values added through the graphical item registration process.
ABPP	Actual bits-Per-Pixel Per Band This field shall contain the number of "valid bits" for the value in each band of each pixel without compression	R/PVV	N/2	01-96
PJUST	Pixel Justification When ABPP is not equal to NBPP, this field indicates whether the most significant bits are left justified (L) or right justified (R).	R/PVV	A/1	R or L

Table 3 -- Image subheader *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
ICORDS	Image Coordinate Type This field shall contain a valid code indicating the coordinate system for the image.	R/PVV	A/1	“space” = None Other codes as defined in profile indicate the existence of the conditional field IGEOLO.
IGEOLO	Image Location utilized for representation of the coordinates of the image. This field is omitted if ICORDS=“space”.	C/PVU	A/60	As defined in profile
NICOM	Number of Image Comments This field shall contain the valid number of 80 character blocks (ICOMn) that follow to be used as free text image comments.	R/PVV	N/1	0-9
ICOM1	Image Comment 1 This field, when present, shall contain free-form BCS-A text. This is the first comment field.	C/PVU	U8/80	
...				
ICOMn	Image Comment n This field, when present, shall contain free-form UTF-8 text. Continuation of the previous comment fields.	C/PVU	U8/80	
IC	Image Compression This field contains a valid profile defined code indicating the form of compression used in representing the image data.	R/PVV	A/2	“NC”=Non-Compressed “NM”=Non-Compressed with mask table “C4”= Vector Quantization Compressed, “M4”=Vector Quantization Masked Additional codes defined in profile
COMRAT	Compression Rate Information This field shall be present and contain a code as defined by profile indicating the compression rate for the image. If the value in IC is C4 or M4, this field shall contain a value given in the form of nn.n representing the number of bits-per-pixel for the compressed image. Explanation of the compression rate for vector quantization can be found in Annex B.	C/PVU	A/4	Field is omitted when “IC”=“NC” or “NM”. As defined in profile for other IC Codes.
NBANDS	Number of Bands This field shall contain the number of bands comprising the image. This field and the IREP field are interrelated and independent of the IMODE field.	R/PVV	A/1	“0”=Number of bands contained in the conditional XBANDS field. “1”-”9” Number of bands “T” Band definition is contained in TFS DES associated with the image segment in the overflow fields.

Table 3 -- Image subheader *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
XBANDS	Number of Multi-Spectral Bands When NBANDS contains the value 0, this field shall contain the number of bands comprising the multi-spectral images with greater than 9 bands. Otherwise this field is empty.	C/PVV	N/5	00010-99999.

NOTE: The fields IREPBANDn through LUTDnm repeat the number of times indicated in the NBANDS field or the XBANDS field. Sequence continuation is indicated elsewhere by a series of dots (…).

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
IREPBAND1	1st Band Representation When NBANDS contains the value one, this field shall contain all spaces. This indicates that the image representation for the single band is that contained in the IREP field. In all other cases, this field shall contain a valid indicator of the interpretation of the first band correlated with the value of IREP as defined in the profile.	R/PVV	A/2	(Default is Spaces) All cases listed in the IREP field are represented. The row below outlines the appropriate values. Additional values are specified through graphical item registration.

IREP	IREPBAND1	IREPBAND2	IREPBAND3	IREPBAND4	IREPBAND5	...
MONO	2 SPACES	N/A	N/A	N/A	N/A	...
RGB/LUT	2 SPACES	N/A	N/A	N/A	N/A	...
RGB	R	G	B	N/A	N/A	...
HIS	H	I	S	N/A	N/A	...
CMY	C	M	Y	N/A	N/A	...
CMYK	C	M	Y	K	N/A	...
YIQ	Y	I	Q	N/A	N/A	...
YUV	Y	U	V	N/A	N/A	...
YCbCr	Y	Cb	Cr	N/A	N/A	...
CIE	C	I	E	N/A	N/A	...
1D	2 SPACES	N/A	N/A	N/A	N/A	...
2D	LX	LY	N/A	NA/	N/A	...
ND	*	*	*	*	*	...
MULTI	1	2	3	4	5	...

* Defined by Profile

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
ISUBCAT1	1st Band Significance for Image Category The use of this field is profile-defined. Its purpose is to provide the significance of the first band of the image with regard to the specific category, ICAT, of the overall image. An example would be the wavelength of IR imagery.	R/PVU	A/6	Profile Defined (Default is Spaces)
IFC1	1st Band Image Filter Condition Profile defined flag of first band filter	R/PVU	A/1	Profile Defined. Default is "N" (None).
IMFLT1	1st Band Standard Image Filter Code Profile defined indicator of first band filter.	R/PVU	A/3	Profile Defined. Default is spaces.
NLUTS1	1st Band Number of LUTS This field shall contain the number of look-up tables associated with the 1st band of the image correlated with IREP.	R/PVV	N/1	"0" - "4"
NELUT1	1st Band Number of LUT Entries This field shall contain the number of entries in each of the look-up tables for the first band of data. This field shall be omitted if the value in NLUTS is zero.	C/PVV	N/5	"00001" - "65536"

Table 3 -- Image subheader *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
LUTD11	Data of 1st LUT for 1st Band This field shall be omitted if the first Band Number of LUTs is zero. Otherwise, this field shall contain the data defining the first look-up table for the first image band. This field supports only integer band data (PVTYPE=INT). Multiple LUTs may be used to translate the index value into multiple octet values.	C	Derived from Value NELUT1	Each entry in the look-up table is composed of one octet.
…..				
LUTD1m	Data of mth LUT for 1st Band This field shall be omitted if the first Band Number of LUTs is zero. Otherwise, this field shall contain the data defining the mth look-up table for the first image band.	C	Derived from Value NELUT1	Each entry in the look-up table is composed of one octet
.....				
IREPBANDn	nth Band Representation This field shall contain a valid indicator of the interpretation of the nth band correlated with the value of IREP. Possible values for n are 00001 to 99999.	C/PVV	A/2	See IREPBAND1.
ISUBCATn	nth Band Sub Category The use of this field is profile-defined. Its purpose is to provide the significance of the nth band of the image with regard to the specific category, ICAT, of the overall image. An example would be the wavelength of IR imagery. Possible values for n are 00001 to 99999.	C/PVU	A/6	Profile Defined
IFCn	nth Band Image Filter Condition Profile defined flag of nth band filter. Possible values for n are 00001 to 99999.	C/PVU	A/1	Profile Defined. Default is “N” (None).
IMFLTn	nth Band Standard Image Filter Code Profile defined indicator of first band filter. Possible values for n are 00001 to 99999.	C/PVU	A/3	Profile Defined. Default is spaces.
NLUTSn	nth Band Number of LUTS This field shall contain the number of look-up tables associated with the nth band of the image correlated with IREP. Possible values for n are 00001 to 99999.	C	N/1	“0”-“4”

Table 3 -- Image subheader *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
NELUTn	n^{th} Band Number of LUT Entries This field shall contain the number of entries in each of the look-up tables for the n^{th} band of data. This field shall be omitted if the value in NLUTS is zero. Possible values for n are 00001 to 99999	C	N/5	“00001”-“65536”
LUTDn1	This field shall be omitted if the n^{th} Band Number of LUTS is zero. Otherwise, this field shall contain the data defining the first look-up table for the nth image band. This field supports only INT band data (PVTYPE = INT). Possible values for n are 00001 to 99999 Multiple LUTs may be used to translate the index value into multiple octet values.	C/PVU	Derived from value of NELUTn	Each entry in the look-up table is composed of one or two octets depending on the value of NBPP. For NBPP<=8; one octet. For 9<=NBPP<=16; two octets. The values of ABPP and PJUST identify the significant bits and bit justification in the table.
.....				
LUTDnm	This field shall be omitted if the n^{th} Band Number of LUTs is zero. Otherwise, this field shall contain the data defining the m^{th} look-up table for the n^{th} image band. Each entry in the look-up table is composed of one octet, ordered from most significant bit to least significant bit representing a value from 0 to 255. This field supports only INT band data (PVTYPE=INT). Possible values for n are 00001 to 99999	C	Derived from value of NELUTn	Each entry in the look-up table is composed of one or two octets depending on the value of NBPP. For NBPP<=8; one octet. For 9<=NBPP<=16; two octets. The values of ABPP and PJUST identify significant bits and bit justification in the table.
ISYNC	Image octet Alignment Code For non-compressed messages this field contains an indicator if end of row or column markers are used.	R/PVV	A/1	Default is “0” indicating no sync codes. Other values may be profile defined.
IMODE	Image Mode This field shall contain an indicator of whether the image bands are stored in the file sequentially or interleaved (by block, row or pixel).	R/PVV	A/1	B=Block Interleaved, P=Pixel Interleaved, S=Band Sequential, R=Row Interleaved.
NBPR	Number of Blocks Per Row This field shall contain the number of image blocks in a row of blocks (see 4.2.5.1) in the horizontal direction. If the image consists of only a single block, this field shall contain the value one.	R/PVV	N/4	“0001”-“9999”; or as constrained by Profile.
NBPC	Number of Blocks Per Column This field shall contain the number of image blocks in a column of blocks (see 4.2.5.1) in the vertical direction. If the image consists of only a single block, this field shall contain the value one.	R/PVV	N/4	“0001”-“9999”, or as constrained by Profile.

Table 3 -- Image subheader *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
NPPBH	Number of Pixels Per Block Horizontal This field shall contain the number of pixels horizontally in each block of the image. It shall be the case that NBPR*NPPBV≥NCOLS.	R/PVV	N/4	“0001”-“9999”, or as constrained by Profile.
NPPBV	Number of Pixels Per Block Vertical This field shall contain the number of pixels vertically in each block of the image. It shall be the case that NBPC*NPPB≥NROWS.	R/PVV	N/4	“0001”-“9999”, or as constrained by Profile.
NBPP	Number of Bits Per Pixel Per Band This field shall contain the number of storage bits used for the value for each component of a pixel vector. The value in this field always shall be greater than or equal to Actual Bits Per Pixel (ABPP). For example, if 11-bit pixels are stored in 16 bits, this field shall contain 16 and Actual Bits Per Pixel shall contain 11.	R/PVV	N/2	“01”-“96”, or as constrained by Profile.
IDLVL	Display level This field shall contain a valid value that indicates the graphic display level of the image relative to other associated BIIF segments in a composite display. The display level of each BIIF segment (image, or symbol) within the BIIF file shall be unique. The image, symbol, or segment in the BIIF file having the minimum display level shall have attachment level zero (unattached).	R	N/3	"001"-"999"
IALVL	Attachment Level This field shall contain a valid value that indicates the attachment level of the image. Valid values for this field are 000 and the display level value of any other image or symbol in the BIIF file. The image or symbol segment in the BIIF file having the minimum display level shall have attachment level zero.	R	N/3	"000"-"998"
ILOC	Image Location The image location is specified by specifying the location of the first pixel of the first line of the image. This field shall contain the image location offset from the ILOC or SLOC value of the segment to which the image is attached or from the origin of the CCS when the image is unattached (IALVL=000). A row or column value of 0 indicates no offset. Positive row and column values indicate offsets down and to the right, while negative row and column values indicate offsets up and to the left.	R	N/10	”RRRRRCCCCC" where for positive row and column values RRRRR= "00000"-"99999" and CCCCC= "00000"-"99999"; for negative row and column values RRRRR= “-0001” - “-9999” and CCCCC= “-0001”-“-9999”

Table 3 -- Image subheader *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
IMAG	Image Magnification Approximation This field shall contain the magnification (or reduction) factor of the image relative to the original source image. Decimal values are used to indicate magnification. Either decimal values or fractions may be used to indicate reduction. Fractions must be a slash (/) followed by an integer (with an implied preceding one). For example: “2.3” and “2.30” both indicate that the original image has been magnified by a factor of 2.3; while “0.5”, “.5”, “.50”, “.500”, and “/2” all indicate that the image has been reduced by half.	R	A/4	.001-9999 /1-/999 Value shall be a fraction (a slash followed by an integer) or a decimal value. Field shall be filled to the right with spaces. The default value is “1.0”
UDIDL	User Defined Image Data Length This field shall contain the length in octets of the entire UDID field plus three (3) octets. The length is three (3) plus the sum of the lengths of all the Tagged Record Extensions (TREs) appearing in the UDID field, since they are not separated from one another. A value of zero shall mean that no Tagged Record Extensions are included in the UDID field. If a Tagged Record Extension is too long to fit in the UDID field, it may be put in a Data Extension Segment (DES). See Section 4.2.8 on Tagged Record Extensions.	R/PVV	N/5	“00000” or "00003"-"99999"
UDOFL	User Defined Overflow If present, this field shall contain “000” if the Tagged Record Extensions in UDID do not overflow into a DES, or shall contain the sequence number of the DES into which they overflow. This field shall be omitted if the field UDIDL contains zero	C	N/3	”000”-”999”
UDID	User defined Image tagged Data If present, this field shall contain Tagged Record Extensions as allowed by profile. The length of this field shall be the length specified by the field UDIDL minus three (3) octets. Tagged record extensions in this field for an image shall contain information pertaining specifically to the image. The TREs shall appear one after the other with no intervening octets. This field shall be omitted if the field UDIDL contains zero.	C/PVU	Tagged Record Extension - Length is specified by UDIDL minus three (3)	TREs as allowed by profile

Table 3 -- Image subheader *(concluded)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
IXSHDL	Extended Subheader Data Length This field shall contain the length in octets of the entire field IXSHD plus three (3) octets. The length is three (3) plus the sum of the length of all Tagged Record Extensions appearing in the IXSHD field, since they are not separated from one another. A value of zero shall mean that no TREs are included in the IXHD field. If a TRE is too long to fit in the IXSHDL field, it shall be put in a Data Extension Segment (DES).	R/PVV	N/5	"00000" or “00003”-"99999"
IXSOFL	Extended Subheader Overflow If present, this field shall contain “000” if the Tagged Record Extensions in IXSHD do not overflow into a DES, or shall contain the sequence number of the DES into which they do overflow. The IXSOFL field shall be omitted if the field IXSHDL contains value of zero.	C	N/3	“000”-”999”
IXSHD	Extended Subheader Data If present, this field shall contain Tagged Record Extensions as allowed by the profile. The length of this field shall be the length specified by the field IXSHDL minus three octets. Tagged record extensions shall appear one after the other with no intervening octets. This field shall be omitted if the field XHDL contains zero.	C/PVU	Tagged Record Extensions length specified by IXSHDL minus three(3)	TREs as allowed by profile

4.2.5 Image data field format

4.2.5.1 Blocked images

Image blocking extends the image model for BIIF presented above to support the representation of an image in terms of an orderly set of subimages (or subarrays) called blocks. A blocked image is analogous to a rectangular tiled floor; regard the overall floor as the image and each individual tile as a block. The performance of an imagery implementation can be potentially improved by “blocking” large images; that is, ordering the pixel values in the file as a series of concatenated pixel arrays (see Figure 9).

4.2.5.1.1 Block size

The number of rows and columns of pixels in each block must be less than or equal to the number of rows and columns of pixels in the overall image. Each block within an image array shall be the same size.

4.2.5.1.2 Block ordering

For recording purposes the image blocks are ordered sequentially by row (of blocks). In Figure 9, the recording order would be B0,0; B0,1; B0,2; B0,3; B1,0; B1,1; B1,2; B1,3; B2,0; B2,1; B2,2; B2,3. The first pixel of each succeeding block immediately follows the last pixel of the preceding block. Although the pixel values are placed in the file as a series of arrays (blocks), the coordinate used to reference any specific pixel remains the same as if the image were not blocked. For example, if each of the blocks in Figure 9 have NPPBV=NPPH=1024, the second pixel in B(0,1) has the coordinate (0,1025) vice the internal index (0,1) of the subarray.

4.2.5.1.3 Block divisibility condition

If the size of the overall image is not an integer multiple of the number of rows or columns in the image, the image shall be padded to an appropriate number of rows and columns so the divisibility condition is met by adding rows or columns to the bottom and right of the image (see Figure 9a and Figure 9b).

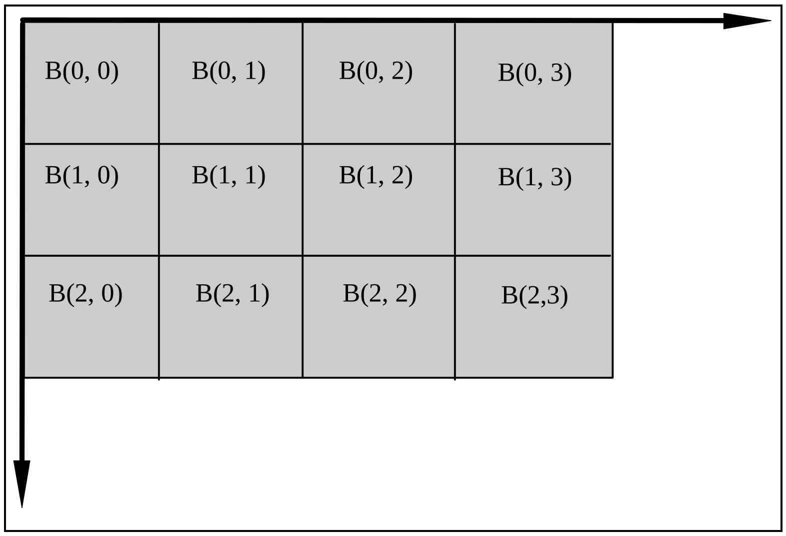


Figure 9a -- A blocked image

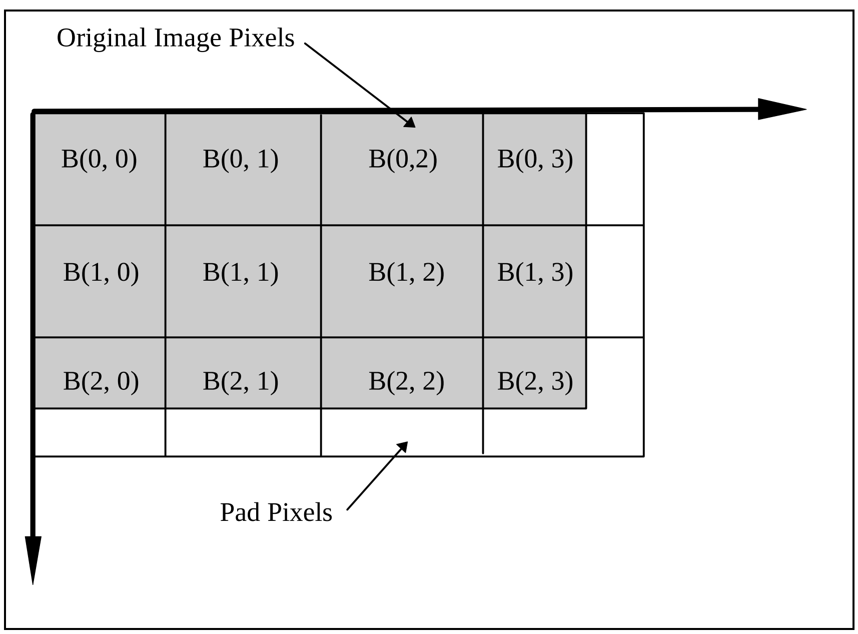


Figure 9b -- A blocked padded image

4.2.5.2 Image data masking

BIIF provides an option to include a blocked image mask table and a pad pixel mask table as part of the image data field immediately following the last octet of the image subheader. The presence of these tables is signalled when the value in the IC field contains the character “M”.

4.2.5.2.1 Blocked image masking

In some instances, a blocked image may have a considerable number of empty blocks. This might occur when a rectangular image is not north aligned, but has been rotated to a north up orientation (see Figure 10). In this case, it is sometimes useful to not record or transmit empty blocks. However, if empty blocks are not recorded/transmitted, the image loses its logical structure as an image with n x m blocks. In order to preclude the loss of logical structure and to allow the exclusion of empty blocks, an image data mask table structure has been defined. The image data mask table is defined in Table 4. The mask identifies the location of non-empty blocks so that the application can reconstruct the image correctly. In Figure 10, the recording order would be B0,0; B0,1; B0,2; B1,0; B1,1; B1,2; B1,3; B2,0; B2,1; B2,2; B2,3; B3,1; B3,2; B3,3. The blocked image mask would identify the locations of the recorded image blocks. If the image is band sequential (IMODE=S), there will be multiple block image masks--one for the each image band, with each mask containing NBPR x NBPC records. Block image masks can be used in conjunction with a pad pixel mask, as described below. A block image mask may also be used to provide random access within the blocked image data for large images even if all blocks are recorded.

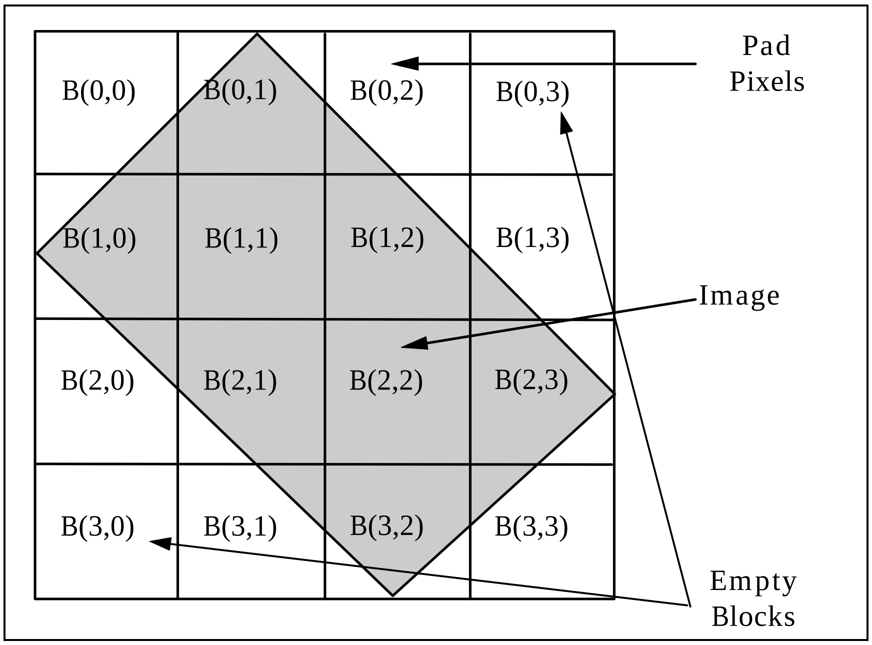


Figure 10 -- A blocked padded image with empty blocks

4.2.5.2.2 Pad pixel masking

In addition to empty image blocks, Figure 10 also demonstrates that a significant number of pad pixels may be needed to "fill" an image to the nearest block boundary. In the example in Figure 10, the locations of image blocks B0,0, B0,1; B0,2; B1,0; B1,2, B1,3, B2,0, B2,1, B2,3, B3,1, B3,2, and B3,3 would be recorded, indicating that these blocks have pad pixels. B0,3; B1,1; B2,2; and B3,0 do not have pad pixels because B0,3 and B3,0 are empty and B1,1 and B2,2 are full image blocks. If the image is band sequential (IMODE=S), there will be pixel masks that will be arranged in the same order as the image bands, with each mask containing NBPR x NBPC records. The output pixel code which represents pad pixels is identified within the Image Data Mask table by the Pad Output Pixel Code field (TPXCD). The length in bits of this code is identified in the Output Pixel Code Length field (TPXCDLNTH). Although this length is given in bits, the actual TPXCD value is stored in an integral number of octets. When the number of bits used by the code is less than the number available in the TPXCD field (for example, a 12 bit code stored in two octets), then the code will be justified in accordance with the PJUST field in the Image Subheader. When an application identifies pad pixels, it may replace them with a user defined value (for example, a light blue background) at the time of presentation except when the value of TPXCD is zero (0). When the TPXCD value is zero, the pad pixel will be treated as "transparent" for presentation. The application may also choose to ignore pad pixels in histogram generation. In any case, pad pixels are not valid data, and should not be used for interpretation. Consequently, the value used for pad pixels in a masked image shall not appear within the bounds of significant pixel values of the image.

4.2.5.2.3 Image data mask table

The image data mask table is a conditional data structure included in the image data stream for masked images (IC value contains an M). The image data mask table is not recorded for non-masked images (IC value without an M). When an image mask table is used, the first octet of the image data is offset from the beginning of the image data area by the length of the image data mask table; and empty image blocks are not recorded/transmitted in the image data area. If the image is band sequential (IMODE=S), there will be multiple block image and/or pad pixel masks--one for each band. All block image masks will be recorded first, followed by all pad pixel masks. Since the image data mask tables are in the image area, the data recorded/transmitted there are binary. The structure of the image data mask table is defined in detail in Table 4.

Table 4 -- Image data mask table

FIELD	Name	TYPE	SIZE-Octets	VALUE RANGE
IMDATOFF	Blocked Image Data Offset - This field is included if the IC value contains the character M. It identifies the offset from the beginning of the Image Data Mask Table to the first octet of the blocked image data. This offset, when used in combination with the offsets provided in the BMR fields, can provide random access to any recorded image block in any image band.	C	4	Unsigned Integer: 0 to 2^{32} - 1 (0x00000000-0xFFFFFFFF)
BMRLNTH	Block Mask Record Length - This field is included if the IC value contains the character M. It identifies the length of each Block Mask Record in octets. The total length of the Block Mask Records is equal to BMRLNTH x NBPR x NBPC x NBANDS. If all of the image blocks are recorded, this value may be set to 0, and the conditional BMR fields are not recorded/ transmitted. If this field is present, but coded as 0, then a pad pixel mask is included.	C	2	Unsigned Integer; 0=No Block mask; 4=Block mask present with each record length is 4 octets (0=0x0000) (4=0x0004)
TMRLNTH	Pad Pixel Mask Record Length This field is included if the IC value contains the character M. It identifies the length of each Pad Pixel Mask Record in octets. The total length of the Pad Pixel Mask Records is equal to TMRLNTH x NBPR x NBPC x NBANDS. If none of the image blocks contain pad pixels, this value is set to 0, and the conditional TMR fields are not recorded/ transmitted. For IC value of M3, the value is set to 0. If this field is present, but coded as 0, then a Block Mask is included.	C	2	Unsigned Integer; 0=No pad pixel mask; 4=Pad pixel mask present , each record length is 4 octets (0=0x0000) (4=0x0004)
TPXCDLNTH	Pad Output Pixel Code Length This field is included if the IC value contains the character M. It identifies the length in bits of the Pad Output Pixel Code. If coded as 0, then no pad pixels are present, and the TPXCD field is not recorded. The length of the TPXCD field is next highest number of octets which can contain the number of bits identified in the TPXCDLNTH field. For example, a TPXCDLNTH value of 12 would be stored in a TPXCD field of two octets.	C	2	Unsigned Integer; 0=No pad pixels; or Pad pixel code length in bits

Table 4 -- Image data mask table *(continued)*

FIELD	Name	TYPE	SIZE-Octets	VALUE RANGE
TPXCD	Pad Output Pixel Code - This field is included if the IC value contains the character M, and TPXCDLNTH is not 0. It contains the output pixel code that represents a pad pixel in the image. This value is unique within the image, and allows the user to identify pad pixels. The pad pixel output code length is determined by TPXCDLNTH, but the value is stored in a maximum of two octets. If the number of bits used by TPXCD is less than the number of bits available for storage, the value shall be justified in accordance with the PJUST field in the image subheader.	C	The length of the TPXCD field is the next highest number of octets which can contain the number of bits identified in the TPXCDLNTH field. (1 or 2 Octets)	Unsigned Integer; 0 to 2^n - 1 where n=TPXCDLNTH
BMR0BND1	Block Mask Record 0. Band 1 This field shall contain the first Block Mask Record of band 1. It is recorded/transmitted only if the BMRLNTH field is not 0. The field shall contain an offset in octets from the beginning of the Blocked Image Data to the first octet of block 0 of band 1 (this value should be 0) if block 0 is recorded/transmitted, or 0xFFFFFFFF if block 0 of band 1 is not recorded/transmitted in the image data.	C	4	Unsigned Integer; Offset in octets from the beginning of Blocked Image Data to the first octet of block 0 of band 1 (usually 0); 0xFFFFFFFF if the block is not recorded.
....				
BMRnBND1	Block Mask Record n, Band 1 This field shall contain the n^{th} Block Mask Record of band 1. It is recorded/transmitted only if the BMRLNTH field is not 0. The field shall contain an offset in octets from the beginning of the blocked Image Data to the first octet of block n of band 1 if block n is recorded or transmitted, or 0xFFFFFFFF if block n of band 1 is not recorded or transmitted in the image data. The number of BMR records for this band is NBPR x NBPC.	C	4	Unsigned Integer; Offset in octets from the beginning of Blocked Image Data to the first octet of block nn of band 1; 0xFFFFFFFF if the block is not recorded.
....				

Table 4 -- Image data mask table *(continued)*

FIELD	Name	TYPE	SIZE-Octets	VALUE RANGE
BMR0BNDm	Block Mask Record 0, Band m This field shall contain the nth Block Mask Record of band m. It is recorded/transmitted only if the BMRLNTH field is not 0. The field shall contain an offset in octets from the beginning of the Blocked Image Data to the first octet of block 0 of band m if block 0 of band m is recorded or transmitted, or 0xFFFFFFFF if block 0 of band m is not recorded or transmitted in the image data. The number of BMR records for this band is NBPR x NBPC. Possible values for m are 00001 to 99999	C	4	Unsigned Integer; Offset in octets from the beginning of Blocked Image Data to the first octet of block 0 of band m; 0xFFFFFFFF if the block is not recorded.
....				
BMRnBNDm	Block Mask Record n, Band m This field shall contain the nnth Block Mask Record of band m. It is recorded/transmitted only if the BMRLNTH field is not 0. The field shall contain an offset in octets from the beginning of the Blocked Image Data to the first octet of block n of band m if block n of band m is not recorded or transmitted in the image data. The number of BMR records for this band is NBPR x NBPC. Possible values for m are 00001 to 99999	C	4	Unsigned Integer; Offset in octets from the beginning of Blocked Image Data to the first octet of block n of band m; 0xFFFFFFFF if the block is not recorded.
TMR0BND1	Pad Pixel Mask Record 0, Band 1 This field shall contain the first Pad Pixel Mask Record for band 1. It is recorded/transmitted only if the TMRLNTH field is not 0. The field shall contain an offset in octets from the beginning of the blocked Image Data to the first octet of block 0 of band 1 if block contains pad pixels, or 0xFFFFFFFF to indicate that this block does not contain pad pixels.	C	4	Unsigned Integer; Offset in octets from the beginning of Blocked Image Data to the first octet of block 0 of band 1; 0xFFFFFFFF if the block does not contain pad pixels.
....				
TMRnBND1	Pad Pixel Mask Record n, Band 1 This field shall contain the nth Pad Pixel Mask Record for band 1. It is recorded/transmitted only if the TMRLNTH field is not 0. The field shall contain an offset in octets from the beginning of the Blocked Image Data to the first octet of block n of band 1 if block n contains pad pixels, or 0xFFFFFFFF to indicate that this block does not contain pad pixels. The number of TMR records for band 1 is NBPR x NBPC.	C	4	Unsigned Integer; Offset in octets from the beginning of Blocked Image Data to the first octet of block n of band 1; 0xFFFFFFFF if the block does not contain pad pixels.
....				

Table 4 -- Image data mask table *(concluded)*

FIELD	Name	TYPE	SIZE Octets	VALUE RANGE
TMR0BNDm	Pad Pixel Mask Record 0, Band m This field shall contain the first Pad Pixel Mask Record for band m. It is recorded/transmitted only if the TMRLNTH field is not 0. The field shall contain and offset in octets from the beginning of the Blocked Image Data to the first octet of block 0 of band m if block 0 contains pad pixels, or 0xFFFFFFFF to indicate that this block does not contain pad pixels. Possible values for m are 00001 to 99999	C	4	Unsigned Integer; Offset in octets from the beginning of Blocked Image Data to the first octet of block 0 of band m; 0xFFFFFFFF if the block does not contain pad pixels.
....				
TMRnBNDm	Pad Pixel Mask Record n, Band m This field shall contain the nth Pad Pixel Mask Record for band m. It is recorded or transmitted only if the TMRLNTH field is not 0. The field shall contain an offset in octets from the beginning of the Blocked Image Data to the first octet of block n of band m if block n contains pad pixels, or 0xFFFFFFFF to indicate that this block does not contain pad pixels. The number of TMR records for band m is NBPR x NBPC. Possible values for m are 00001 to 99999.	C	4	Unsigned Integer; Offset in octets from the beginning of Blocked Image Data to the first octet of block n of band m; 0xFFFFFFFF if the block does not contain pad pixels.

4.2.5.3 Compressed image data format

The format of the image data placed in the image data field after compression is provided with the description of BIIF image compression algorithms in the appropriate Standards Documents selected by the profile (see Clause 2).

4.2.5.4 Uncompressed image data format

The order in which pixel values of a single band image are stored is fixed. When an image has more than one band, several options are available for the order in which pixel values are stored. The option used is indicated by the IMODE field in the image subheader. The following subclauses describe the possibilities within this format. In describing the encoding of image data. The BIIF display convention is invoked freely for ease of expression. Let the image to be encoded be denoted by I, and assume I has R rows and C columns. Let I have n bands; that is, each pixel is an n-vector, the i^{th} value of which is the value for that pixel location of the i^{th} band of the image. Let N denote the number of bits-per-pixel-per-band. Thus, there are n*N bits-per-pixel. Let I be blocked with H blocks per row and V blocks per column. Note that special cases such as single band images and single block images are included in this general image by setting n=1, and H=V=1, respectively.

4.2.5.4.1 Single band image uncompressed data format

For single band images, where n=1, and there is only one order for storing pixels. The field IMODE in the image subheader shall be set to B for this case. The blocks (one or more) shall be stored, one after the other, starting with the upper left block and proceeding first left to right across rows of blocks, one row of blocks after the other, top to bottom. Image data within each block shall be encoded as one continuous bit stream, one pixel value after another, beginning with the N bits of the upper left corner pixel, I(0,0), followed by the N bits of I(0,1) and so on until all pixels from the first row in the block are encoded. These shall be followed immediately by the N bits of data for pixel I(1,0) continuing from left to right along each row, one row after another from the top of the block to the bottom. The last octet of each block's data is zero-filled (if necessary) to the next octet boundary, but all other octet boundaries within the block are ignored. See Table 2 for the specification of the bit representation of pixel values.

4.2.5.4.2 Multiple band image uncompressed data format

For multiple band images, there are four orders for storing pixels.

4.2.5.4.2.1 Band sequential

The first case is "band sequential", in which each band is stored contiguously, starting with the first band, one after the other, until the last band is stored. Within each band the data shall be encoded as if it were a single band image with multiple blocks. The field IMODE in the image subheader shall be set to S for this case. The IMODE S is only valid for images with multiple blocks and multiple bands. (For single block images, this case collapses to the “band interleave by block” case, where IMODE is set to B.)

4.2.5.4.2.2 Band interleaved by pixel

The ordering mechanism for this case stores the pixels in a block sequential order in which each block is stored contiguously, starting with the upper left block and proceeding first left to right across rows of blocks, one row of blocks after the other, top to bottom. Within each block, the n*N bits of the entire pixel vector are stored pixel-by-pixel in the same left to right, top to bottom pixel order as described in 4.1.5.5.2.1. The n*N bits for a single pixel are stored successively in this order: the N bits of the first band followed by the N bits of the second band and, so forth, ending with the N bits of the last band. Each block shall be zero-filled to the next octet boundary when necessary. The field IMODE in the image subheader shall be set to P for this storage option. See Table 2 for the specification of the bit representation of pixel values for each band.

4.2.5.4.2.3 Band interleaved by block

The ordering mechanism for this case stores the pixels in a block sequential order where each block is stored contiguously, starting with upper left block and proceeding first left to right across rows of blocks, one row of blocks after the other, top to bottom. Within each block, the data from each band is stored starting with the first band, one after the other until the last band is stored. Each block shall be zero-filled to the next octet boundary when necessary. The field IMODE in the image subheader shall be set to B for this storage option. See Table 2 for the specification of the bit representation of pixel values for each band.

4.2.5.4.2.4 Band interleaved by row

The ordering mechanism for this case stores the pixel values of each band in row sequential order. Within each block, all pixel values of the first row of the first band are followed by pixel values of the first row of the second band continuing until all values of the first row are stored. The remaining rows are stored in a similar fashion until the last row of values has been stored. Each block shall be zero filled to the next octet boundary when necessary. The field IMODE shall be set to R for this option.

4.2.5.5 Vector quantized data

Vector quantization is a structuring mechanism for use on multiband, colour, and gray scale, scanned maps and imagery. The fundamental concept of vector quantization is to represent the image using a clustering technique to develop a codebook of quantized values. The indices of the codebook replace the image data in the BIIF file. The decoding mechanism is fully discussed in Annex D of this document.

4.2.5.6 Number of bands (NBANDS)

The NBANDS field in the image subheader shall define how many bands of imagery data are included in the BIIF file. If the value of this field is 1-9, the bands are homogeneous and are processed identically in accordance with parameters defined in the image subheader. For images greater than 9 bands, the NBANDS field is set to zero(0) and the number of bands is represented in the conditional field, XBANDS. When bands are not homogeneous (e.g., mix of spectral and time values) the value of NBANDS is set to T, and the image band parameters are expanded by the Data Extension Segment described in the Transportable File Structure, Annex A, for use of PIKS object classes (e.g., PIKS_IMAGE).

4.2.6 Symbol segment

4.2.6.1 Symbol subheader

Each symbol segment has a symbol subheader immediately preceding the symbol data. The symbol subheader is used to identify and supply the information necessary to display the symbol data as intended. The format and description for the symbol subheader is shown in Table 5.

Table 5 -- Symbol subheader

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
SY	BIIF Symbol Subheader This field shall contain the characters SY to identify the subheader as a symbol subheader.	R	A/2	"SY"
SID	Symbol ID This field contains an identification associated with the symbol.	R/PVU	A/10	Profile Defined
SNAME	Symbol Name This field shall contain a name for the symbol.	R/PVU	U8/20	Profile Defined Default is spaces.
SSSEC	Symbol Security Profile Specific Parameters This field shall contain profile specific information for the symbol segment security.	R/PVU	U8/167	Profile Defined
ENCRYP	Encryption Encryption codes and meaning can be defined by profile.	R/PVU	A/1	0=not encrypted. Other codes as specified in profile.
SFMT	Symbol Format: This field shall contain a valid indicator of the representation type of the symbol. A valid value is C, which means Computer Graphics Metafile (CGM). This field identifies the standard as being formatted according the specified standard. Additional symbol format values may be added through the graphical item registration process.	R/PVV	A/1	“C”=CGM. Other codes specified in profile
SSTRUCT	Symbol structure parameters This field shall contain profile defined information describing the type and structure of the data (e.g. CGM clear text encoding) in the symbol data field.	R/PVU	A/13	Profile Defined

Table 5 -- Symbol subheader *(continued)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
SDLVL	Display level This field shall contain a valid value that indicates the graphic display level of the component relative to other BIIF segment. The display level of each BIIF segment (image or symbol) within the BIIF file shall be unique. The image or symbol segment in the BIIF file having the minimum display level shall have attachment level zero.	R	N/3	"001"-"999"
SALVL	Attachment Level This field shall contain a valid value that indicates the attachment level of the component. Valid values for this field are 000 and the display level value of any other image or symbol in the BIIF file. The image or symbol component in the BIIF file having the minimum display level shall have attachment level zero.	R	N/3	"000"-"998"
SLOC	Symbol Location The symbol location is specified by providing the location of the symbol's origin point relative to the position (location) of the item to which it is attached. This field shall contain the symbol location offset from the ILOC or SLOC value of the item to which the symbol is attached or from the origin of the CCS when the symbol is unattached (SALVL=000). A row or column value of 000 indicates no offset. Positive row and column values indicate offsets down and to the right, while negative row and column values indicate offsets up and to the left.	R	N/10	"RRRRRCCCCC" where for positive row and column values RRRRR= "00000"-"99999" and CCCCC= "00000"-"99999"; for negative row and column values RRRRR= "-0001" - "-9999" and CCCCC= "-0001"-"-9999"
SLOC2	Second Symbol Location Default is all zeroes.	R/PVU	N/10	Profile defined.
SPARMS	Symbol profile specific parameters This field shall contain profile specific information for the symbol such as colour, rotation angle, etc.	R/PVU	A/13	Profile Defined
SXSHDL	Extended Subheader Data Length This field shall contain the length in octets of the entire SXSHD field plus three (3) octets. The length is three plus the sum of the lengths of all the Tagged Record Extensions appearing in the SXSHD field. A value of zero shall mean that no Tagged Record Extensions are included in the symbol subheader. If a Tagged Record Extension is too long to fit in the SXSHD field, it shall be put in a data extension segment.	R/PVV	N/5	"00000" or "00003" - "09741"

Table 5 -- Symbol subheader *(concluded)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
SXSOFL	Extended Subheader Overflow If present, this field shall contain “000” if the Tagged Record Extensions in SXSHD do not overflow into a DES, or shall contain the sequence number in the file of the DES into which they do overflow. This 3 octet field must be included in the total octet count of SXSHDL. This field shall be omitted if the field SXSHDL contains zero.	C	N/3	000-999
SXSHD	Extended Subheader Data If present, this field shall contain Tagged Record Extensions as allowed by the profile. Tagged record extensions in this field symbol shall contain information pertaining specifically to the symbol. Tagged record extensions shall appear one after the other with no intervening octets. This field shall be omitted if the field SXSHDL contains zero.	C/PVU	length specified by SXSHDL minus "3"	Tagged Record Extensions as allowed by Profile

4.2.6.2 Symbol data

In the BIIF file, the information describing a symbol segment is represented in a series of adjacent fields grouped into the symbol subheader followed by the symbol data. The field containing the symbol data is called the symbol data field. The symbol data field shall follow immediately after the last field of the corresponding symbol subheader with no intervening special characters to designate the beginning of the symbol data field. Similarly, the symbol subheader of the first symbol shall follow immediately the last octet of data of the last field in the BIIF image section, and the symbol subheader of successive symbols shall follow immediately the last octet of the symbol data field of the preceding symbol. The format of the symbol data field content is specified by the profile. The code “C” in the STYPE field shall designate the use of ISO/IEC 8632, Computer Graphics Metafile (CGM). The BIIF profile shall designate which ISP (s) of CGM are applicable to the specific profile of BIIF.

4.2.7 Text information segment

The text information segment shall be used to store a textual based file or an item of text, such as a word processing file or document. Text items are intended to convey information about the image product contained in the BIIF file. They are not displayed as part of an image.

4.2.7.1 Text subheader

The text subheaders used to identify and supply the information about the text Information necessary to read and display the text. The format and descriptions for the text subheader is shown in Table 6.

Table 6 -- Text subheader

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
TE	Text Information Subheader This field shall contain the characters "TE" to identify the subheader as a text subheader	R	A/2	"TE"
TEXTID	Text ID This field shall contain a valid identification code associated with the text item.	R/PVU	A/10	Profile Defined
TXTDT	Text Date & Time This field shall contain the UTC (Z) date/time of origination.	R	N/14	CCYYMMDDhhmmss
TXTITL	Text Title This field shall contain the title of the text item.	R/PVU	U8/80	Profile Defined
TSSEC	Text Security Profile Specific Parameters This field shall contain profile specific information for the text product security and is defined in the profile.	R/PVU	U8/167	Profile Defined
ENCRYP	Encryption Encryption codes and meaning can be defined by profile, however, 0 = not encrypted.	R/PVU	A/1	0=not encrypted As specified in profile
TXTFMT	Text Format A three character code indicating the format or template to be used to display the text.	R/PVU	A/3	“STA”. “UC2”, “UC4”, “UT1”, UT8”. Other codes specified by profile.
TXSHDL	Extended Subheader Data Length This field shall contain the length in octets of the entire TXSHD field plus three (3) octets. A value of zero shall mean that no Tagged Record Extensions are included in the text subheader. If a Tagged Record Extension is too long to fit in the TXSHD field, it shall be put in a data extension segment.	R/PVV	N/5	"00000" or “00003” - "09717"
TXSOFL	Extended Subheader Overflow If present, this field shall contain “000” if the Tagged Record Extensions in TXSHD do not overflow into a DES, or shall contain the sequence number in the file of the DES into which they do overflow. This 3 octet field must be included in the total octet count of TXSHDL. This field shall be omitted if the field TXSHDL contains zero.	C	N/3	000-999

Table 6 -- Text subheader *(concluded)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
TXSHD	Extended Subheader Data If present, this field shall contain Tagged Record Extensions as allowed by the profile. The length of this field shall be the length specified by the field TXSHDL plus three (3). Tagged record extensions in this field for text shall contain information pertaining specifically to the text segment. Tagged record extensions shall appear one after the other with no intervening octets. This field shall be omitted if the field TXSHDL contains zero.	C/PVU	- length specified by TXSHDL minus three (3)	Tagged Record Extensions allowed by profile

4.2.7.2 Text

The text data field containing the textual item included in the BIIF file shall follow the corresponding text subheader. The text shall consist entirely of characters permitted by the text format code specified in the subheader. Text may be specified in any allowable ISO 10646 manner as defined by profile. The code "STA" in the TXTFMT field shall mean that only BCS-A characters appear in the text data field. A TXTFMT value of UC2 indicates UCS-2, two octet UCS , UC4 is UCS-4, four octet UCS, UT1 is UTF-1, UCS Transformation Format 1, and UT8 is UTF-8, UCS Transformation Format 8 (Amendment 2 of ISO 10646). Although formatting control characters, CR, LF, FF, are also allowed, these TXTFMT codes indicate that the text characters are otherwise unformatted. Additional codes may be defined by profile to specify formatted text.

4.2.8 Data Extensions

In order to provide complete extensibility to the otherwise fixed format of BIIF, data extensions are available that provide flexibility and versatility. There are three major types of data extensions, Tagged Record Extensions, Date Extension Segments and Reserved Extension Segments.

4.2.8.1 Tagged Record Extensions (TRE): Public and Private.

The Tagged Record Extension is used to extend BIIF by adding additional attributes to existing segments (e.g., IM, SY, TE, see Tables 3, 5, and 6). Although their structure is similar, there are two distinguishing attributes for TREs: the Public TRE and the Private TRE.

Public TREs have identifier names which must be registered by the ISO registration authority. The Private TRE is controlled by the community or group which uses it. All Public TREs have an asterisk "*" in the first location of their identifier name. Private TREs have no asterisk in their name.

The purpose of Private TREs is to allow users of BIIF to establish user defined data constructs within the BIIF structure without the need for international consensus.. The private TRE identifier and the structure and content of the user-defined data (TREDATA field) are not coordinated, controlled, nor configuration managed by the ISO or its designated registration authority. Consequently, the use of the TRE is considered to be private in the sense that a specific TRE may only be meaningful to those BIIF users who have mutually agreed to use the TRE in a specified manner.

The following Public TRE usage concepts apply:

— All Public TRE have an asterisk "*" in the first location of its identifier name. The asterisk in the identifier name is the only indicator that an TRE is Public.
— Only those Public TREs accepted and registered by the registration authority shall be used.
— Public TREs shall not be used nor submitted for registration if they adversely impact the utility of the standard features otherwise defined within BIIF and its extensions.
— Nominated Public TREs will be recorded in the 'Register' upon approval by the registration authority.

— Upon receipt of a file that contains Public TREs, a BIIF compliant system shall at least ignore the TREs and properly interpret the other legal components of BIIF file.

The following Private TRE usage concepts apply:

— All Private TRE do not have an asterisk “*” in the first location of its identifier name.
— Private TREs are not registered by the ISO registration authority.
— Private TREs should be ignored by all disinterested bodies.
— Upon receipt of a file that contains Private TREs, a BIIF compliant system shall at least ignore the TREs and properly interpret the other legal components of BIIF file.

Each TRE consists of three required fields. These fields are defined in Table 7. A sequence of extensions can appear in the BIIF file header User Defined Data field, UDHD, in any image subheader in its User Defined Image Data field (UDID) and in extended header and subheader data fields (XHD, IXSHD, SXSHD, TXSHD). When the extension carries data associated with the BIIF file as a whole, it should appear in the file header, if sufficient room is available. If the extension carries data associated with an image data item in the BIIF file, it should appear in the UDID or IXSHD field of that item's subheader, if sufficient room is available. Overflow for the TRE may appear in a Data Extension Segment (DES) that is designated to contain extensions. A tagged record extension shall be included in its entirety within the header, subheader or DES selected to contain it. See the Informative Annex E for a sample BIIF File Structure.

Table 7 -- TRE subheader

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
TRETAG	Unique extension type identifier This field shall contain a valid tag name defined by the profile	R	A/6	For PUBLIC, 6 BCS-A characters, the first one is an “*”, For PRIVATE, 6 BCS-A characters where the first character shall not be an “*”.
TREL	Length of TREDATA field This field shall contain the length in octets of the data contained in TREDATA	R	N/5	"00001"-"99985"
TREDATA	User or profile defined data This field shall contain data defined by and formatted according to user or profile specification. The length of this field shall not cause any other BIIF field length limits to be exceeded, but is otherwise fully user-defined.	R	As specified by TREL	For PUBLIC Tagged Record Extensions defined through registration process For PRIVATE TREs User Defined

4.2.8.2 Data extension segments (DES)

The Data Extension Segment (DES) construct allows for the addition of a variety of differing data types where each type is encapsulated in its own DES. This means that a DES can carry almost any conceivable data type. Potential data types include such things as augmenting imagery with voice annotations, video clip annotations, video/voice annotations, animated graphics, TRE Overflows, Transportable File Structures (TFSs), etc. Each DES carries only one data type, but multiple DESs may be included in a BIIF file, each with the same or differing data type. Each encapsulated extension shall appear in its own Data Extension Segment (DES) and shall conform to the DES structure, Table 8.,

The following DES usage concepts apply:

— Only those DES(s) accepted and registered by the registration authority shall be used.
— Upon receipt of a file that contains DES(s), a BIIF compliant system shall at least ignore the DES(s) and properly interpret the other legal components of BIIF file.
— Implementations supporting a specific DES shall at least comply with the minimum conformance requirements specified in the DES description.

The BIIF file header accommodates up to 999 DES. Each DES shall consist of a DES subheader and a DES data field. Within the Data Extension Segment Group in the BIIF file header is found the number of DES(s) in the BIIF file, the length of each DES subheader, and length of the DES data field, DESDATA. The field size specifications in the BIIF file header allows each DES to be just less than one gigabyte in length. The DES subheader shall contain the fields defined in Table 8.

Table 8 -- Data extension segment subheader

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
DE	Data Extension Subheader This field shall contain the characters "DE" to identify the subheader as a data extension.	R	A/2	"DE"
DESID	Unique DES type identifier This field shall contain a valid identifier as registered with the registration authority.	R	A/25	“TRE_OVERFLOW”, or "TRANSPORTABLE_FILE _STRUCT". or as allowed by profile
DESVER	Version of the data field definition This field shall contain the version number of the use of the tag as registered with the registration authority.	R	N/2	"01-”99”
DESCLAS	Security Profile Specific Parameters This field shall contain profile specific information for the DES security and is defined in the profile using the DES.	R	U8/167	As specified by the applicable profile using this DES.
DESOFLW	Overflowed header type This field shall be present if DESID = "TRE_OVERFLOW". Its presence indicates that the DES contains a tagged record extension that would not fit in the BIIF file header or component header where it would ordinarily be located. Its value indicates the data type to which the enclosed tagged record is relevant.	C/PVV	A/6	UDHD, UDID; XHD, IXSHD, SXSHD, TXSHD; otherwise, field is omitted.
DESITEM	Data item overflowed This field shall be present if DESOFLW is present. It shall contain the number of the data item in the BIIF file, of the type indicated in DESOFLW to which the tagged record extensions in the segment apply. For example, if DESOFLW = UDID and DESITEM = 3, then the tagged record extensions in the segment applies to the third image in the BIIF file. If the value of DESOFLW = UDHD or XHD, the value of DESITEM shall be 000.	C	N/3	"000"-"999"
DESSHL	Length of DES-defined subheader fields This field shall contain the number of octets in the field DESSHF. If this field contains 0, DESSHF shall not appear in the DES subheader.	R	N/4	"0000"-"9999"

Table 8 -- Data extension segment subheader *(concluded)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
DESSHF	DES-defined subheader fields This field shall contain DES-defined subheader fields. Data in this field shall be formatted according to the registered DES specification.	C/PVU	Value specified in DESSHL	Defined through registration process
DESDATA	DES-defined data field This field shall contain data of either binary or character types defined by and formatted according to the user's specification. However, if the DESID is "TRE_OVERFLOW," the tagged records shall appear according to their definition with no intervening octets. If the DESID is “TRANSPORTABLE_FILE_STRUCT" then the format is specified in Annex A. The length of this field shall not cause any other BIIF field length limits to be exceeded, but is otherwise fully user-defined.	R/PVU	Per value from file header	Defined through registration process

4.2.8.3 Defined DESs

There are two currently defined DES type identifiers: ”TRE_OVERFLOW” and ”TRANSPORTABLE_FILE_STRUCT".

4.2.8.3.1 TRE_OVERFLOW DES

The “TRE_OVERFLOW" is used when a series of TRE extensions appears in a DES as "overflow" from BIIF header or any subheader. A separate DES is used for each header or subheader field that overflows. Which header or subheader field overflowed is indicated in the DESOFLOW and DESITEM field contents. The DES for encapsulating Tagged Record Extensions which overflow from the file header or standard segment subheaders is defined in Table 9.

Table 9 -- Data extension segment subheader for TRE OVERFLOW

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
DE	Data Extension Subheader This field shall contain the characters "DE" to identify the subheader as a data extension.	R	A/2	"DE"
DESID	Unique DES type identifier	R	A/25	“TRE_OVERFLOW”
DESVER	Version of the data field definition This field shall contain the version number of the use of the tag as registered with the registration authority.	R	N/2	"01"
DESCLAS	Security Profile Specific Parameters This field shall contain profile specific information for the DES security. and is defined in the profile using this DES.	R	U8/167	As specified by the applicable profile using this DES

Table 9 -- Data extension segment subheader for TRE_OVERFLOW *(concluded)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
DESOFLW	Overflowed header type This field shall be present if DESTAG = "TRE_OVERFLOW". Its presence indicates that the DES contains a TRE that would not fit in the BIIF file header or component header where it would ordinarily be located. Its value indicates the data type to which the enclosed tagged record is relevant.	C	A/6	UDHD, UDID, XHD, IXSHD, SXSHD, TXSHD; otherwise, field is omitted.
DESITEM	Data item overflowed This field shall be present if DESOFLW is present. It shall contain the number of the data item in the BIIF file, of the type indicated in DESOFLW to which the Tagged Record Extensions in the segment apply. If the value of DESOFLW = UDHD or XHD, the value of DESITEM shall be 000.	C	N/3	"000"-"999"
DESSHL	Length of DES-defined subheader fields	R	N/4	"0000"
DESDATA	DES-defined data field This field shall contain data of either binary or character types defined by and formatted according to the user's specification. The length of this field shall not cause any other BIIF field length limits to be exceeded, but is otherwise fully user-defined.	R	Determined by User, Profile Defined	Tagged Record Extensions with no intervening octets.

4.2.8.3.2 TRANSPORTABLE_FILE_STRUCT DES

The TRANSPORTABLE_FILE_STRUCT DES is used to contain TFS Commands as specified in Annex A. This DES is defined in Table 10.

Table 10 -- Data extension segment subheader for TRANSPORTABLE_FILE_STRUCT

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
DE	Data Extension Subheader	R	A/2	"DE"
DESID	Unique DES type identifier	R	A/25	"TRANSPORTABLE_FILE_STRUCT"
DESVER	Version of the data field definition This field shall contain the version number of the use of the tag as registered with the registration authority.	R	N/2	"01"
DESCLAS	Security Profile Specific Parameters This field shall contain profile specific information for the DES security and is defined by the profile using the DES.	R	U8/167	As specified by the applicable profile using this DES

Table 10 -- Data extension segment subheader for TRE Overflow *(concluded)*

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
DESSHL	Length of DES-defined subheader fields	R	N/4	"0000
DESDATA	DES-defined data field This field shall contain data of either binary or character types defined by and formatted according to the user's specification. The length of this field shall not cause any other BIIF field length limits to be exceeded, but is otherwise fully user-defined.	R	Per value from file header	TFS Commands as specified in Annex A

4.2.8.4 Reserved extension segments (RES)

The RES construct provides the same mechanism as the DES construct for adding a variety of new data types for inclusion in BIIF files. No use of the RES construct is currently specified in this standard. The RES is reserved for data types that need to be placed at or near the end of the file. For example, a digital signature that covered the whole file could be defined for placement in a RES to verify the bit level integrity of BIIF file. The RES subheader shall contain the fields defined in Table 11.

The following RES usage concepts apply:

- Only those RES(s) accepted and registered by the registration authority shall be used.
- Upon receipt of a file that contains RES(s), a BIIF compliant system shall at least ignore the RES(s) and properly interpret the other legal components of BIIF file.
- Implementations supporting a specific RES shall at least comply with the minimum conformance requirements specified in the RES description.

Table 11 -- Reserved extension segment subheader

FIELD	DESCRIPTION	TYPE	CE/SIZE	VALUE RANGE
RE	Reserved Extension Subheader This field shall contain the characters "RE" to identify the subheader as a data extension.	R	A/2	“RE”
RESID	Unique RES type identifier This field shall contain a valid identifier as defined through registration.	R	A/25	Unique Identifier as registered with ISO
RESVER	Version of the data field definition This field shall contain the version number of the use of the tag as defined by the registration process.	R	N/2	"01"-"99"
RESSEC	Security Specific Parameters This field shall contain specific information for the product security as defined in the profile being used.	R	U8/167	As specified for the applicable profile using the RES
RESSHL	Length of registration-defined subheader fields This field shall contain the number of octets in the field RESSHF. If this field contains 0, RESSHF shall not appear in the RES subheader.	R	N/4	"0000"-"9999"
RESSHF	Registration-defined subheader fields This field shall contain registration-defined subheader fields. Data in this field shall be formatted according to registered specification.	C/PVU	Value specified in RESSHL	Defined through registration
RESDATA	Registration-defined data field This field shall contain data of either binary or character types defined by and formatted according to the user's specification.	R/PVU	Per value from file header	Defined through registration

5 Conformance profiles and extensions

ISO/IEC 12087-5 (BIIF) utilizes the concept of conformance profiles as established by ISO/IEC 12087-1 and International Standardized Profiles (ISPs) as established by ISO/IEC TR 10000-1 Third Edition, 1995-12-15. Registering an ISP is the standardized means for tailoring BIIF for use by communities of interest that have different functional scopes in order to suit a variety of user requirements. Since BIIF is very flexible, it has many options; the use of which must be constrained for implementation if file exchange interoperability is to be achieved within a designated community of interest. The BIIF profiling approach allows specification of a hierarchical (nested) set of implementation variables (complexity levels) within a specific ISP. The BIIF profile allows inclusion of data types defined by external content profiles (profiles of ISO/IEC 12087, registered ISO profiles external to 12087, and other approved standards documents and registered items). Finally, the BIIF profile provides additional means of extensibility through the registration of tagged and encapsulated extensions.

5.1 Profiles

A Model Profile is provided as a normative annex to this standard (Annex C). This Model Profile consists of a set of proforma tables to be used as templates for specifying BIIF ISPs.

New ISPs nominated for registration can be developed by referring to the Model Profile as the starting point and simply identifying additional capabilities and their constraints. Although the inclusion of the Model Profile within registered profiles is not mandatory, it is considered a minimally conformant use of BIIF. Inclusion of the Model Profile in other profiles promotes an increased potential for a basic level of interoperability and data portability among implementations of differing BIIF profiles. This basic level of interoperability can be achieved by providing an implementation option which allows the user to limit the content of BIIF files to the constraints of the model profile.

When developing a new profile for a specific application domain, the Model Profile, existing profiles and referenced content profiles (such as those for PIKS, CGM, JPEG, etc.) should be examined to determine if these meet the requirements for the targeted application domain. The Model Profile may be used as the starting point for the new profile, or other profiles that have been developed for communities with similar interests may be used.

5.2 Profile specific header/subheader dependencies proforma

The standard describes a number of header and sub-header fields, the formats of which are subject to profile specific definition and constraints. Table C.1 through C.9 in Annex C provides a proforma to assist in the submission of the formally registered profiles of this standard. A completed proforma, once validated and approved by the registration authority, becomes the definition of profile specific constraints to be used by implementors of the profile.

5.3 Complexity level proforma

Complexity levels within a functional profile provide a means to define a set of nested features and constraints within a specific profile. Having these nested feature sets (where the restrictions on the set of attributes are expected to vary in progression and only for specified values) increases the potential for a high degree of interoperability between disparate implementations having various levels of computational resources (e.g. laptops, desktops, workstations, super-computers, etc).

A "complexity level" is defined by a nested set of Profile Variants (PVs) that are a proven combination that work together. The same set of profile variants is in effect across the range of complexity levels with the general framework of support remaining the same for all.

The rules for developing complexity levels within a single Functional Profile are:

- Select the set of Profile Variants from those permitted by BIIF standard (clause 4 tables indicate attributes that are considered to be potential profile variants);
- Specify the permitted values (minimum to maximum, or from an enumerated set or special syntax and semantics as defined by the profile) for each PV at each complexity level.
- All other data conformance requirements (the "framework") remain the same across the complexity level progression.

The proformas presented in Annex C provide a standard way of expressing the selections.

5.4 Implementation support requirements

There are three primary implementation support considerations in establishing compliance with BIIF:

- Compliance of individual files with the selected BIIF profile;
- Ability to produce BIIF compliant files within profile constraints; and,
- Ability to properly interpret and present BIIF compliant files.

The above clauses describe aspects of BIIF profiling that address the first consideration above, the compliance of the file format itself. This is the primary purpose of this standard. However, the BIIF profiling approach allows for the profile proponent to identify implementation support requirements applicable to the concept of operation within the application domain envisioned to use BIIF profile. There is no proforma specified for inclusion of implementation support requirements as part of a registered profile; the format is free text.

Functional areas to be addressed should include at a minimum, the following:

- Image reproductive fidelity (e.g., colour fidelity, requirements for mensuration, font fidelity of shape and size, level of geometric precision, etc.)
- Compression goals and related quality of image.
- Any special content transforms that may be needed to correctly interpret data (e.g., the mathematical algorithms for converting from a non-standard colour model to a standard model).
- Any special handling or processing rules that provide application semantics (e.g., the special interpretation of a hyperlink, a user interface function, a timing consideration or constraint.).

5.5 Defined extensions

Extensibility of BIIF is supported by the registration and use of Tagged Record Extensions (TREs), Data Extension Segments (DESs), and Reserved Extension Segments (RESs) as defined in Clause 4. The registration process is that defined by ISO/IEC 9973, Procedures for Registration of Graphical Items.

5.6 Profile registration

Alternative profiles of BIIF may be created and registered for general usage. These profiles shall follow the model profile proforma shown in Annex B. The model profile lists all the required components, of which selections, restrictions, and qualifications may be made. These can include restrictions on options and fields.

Profiles may reference standardized portions of the standard, registered extensions of the standard, or other ISO standards and ISPs. Profiles shall not contain additional functionality. Additional content should be registered separately.

The registration process for ISPs generally involves the following steps as detailed in ISO/IEC TR10000-1:

1. Develop the profile per the Model Profile
2. Submit the profile to the Registration Authority
3. Review of the profile by the Registration Authority
4. If the Registration Authority approves, skip to step 6
5. If the Registration authority makes a request for revision, return to step 1
6. International ballot of profile;
7. If there is International approval, the profile becomes available from ISO & Registration Authority
8. If there is International denial, return to step 1.